<http://www.cnblogs.com/jose-Lee/p/4896088.html>

（摘自西西软件园，原文链接http://www.cr173.com/html/46224\_1.html）

解决版本冲突的命令。在冲突解决之后，需要使用svnresolved来告诉subversion冲突解决，这样才能提交更新。冲突发生时，subversion会在WorkCopy中保存所有的目标文件版本（上次更新版本、当前获取的版本，即别人提交的版本、自己更新的版本、目标文件。

开发人员都知道代码管理工具是开发中一个必不可少的工具，这里也不废话详细介绍了。不管你个人喜欢git还是svn还是其他，但还有一大部分公司在使用svn做代码管理工具。这里详细介绍下SVN提交文件时冲突问题的解决方式。

假设A、B两个用户，他们分别从svn服务器中检出了test1.txt文件，此时A、B、服务器三个地方的test1.txt的版本都是13(我测试环境的当前svn赋予的版本号)。A、B文件的内容如下图（左A右B）：

·

接下来，B用户添加一句话并提交，内容如下：

此时B用户和服务器的test1.txt的版本都变为14，只有A用户的test1.txt的版本还为13。接下来A用户添加一句“aa”,然后提交

由于A用户是在13版本上做的修改，而服务器已经是14版本了，所以会提交失败：

接下来就是我们要解决的问题了，解决方法分为以下两种方式。第一种方式：提交失败后直接选择revert，省去了解决冲突问题；第二种方式：提交失败后选择更新文件，这时会有冲突问题。详细介绍如下：

第一种方式：

A放弃自己修改的内容，进行Revert操作，使其test1.txt成为13版本的最初内容。然后update使其test1.txt成为14版本，再在14版本上修改提交。操作如下图：

==》

   ==>然后再修改提交

第二种方式：

因为版本过时，提交失败后。A用户直接选择更新操作，结果如下图所见

（这里声明下，不要被文件显示的图标所迷惑，这是其他软件对它做了关联导致的，没啥影响）

这里详细说一下产生冲突后的这几个文件，：

test1.txt.mine---这个文件是A用户在13版本中做了修改要提交的文件。它的内容是：13版本内容+A用户的修改

test1.txt.r13----这个文件是A用户最初的13版本的test1.txt。它的内容是：13版本内容

test1.txt.r14----这个文件时svn服务器中test1.txt的最新版本，这里既是B用户提交后的14版本。它的内容是：13版本内容+B用户的修改

test1.txt--------由于A用户选择了直接更新，此文件就是svn将 最新版本14 与 A用户的修改 合并后的文件。它的内容如下：

接下来说一下如何解决。对于源代码文件或其他的纯文本文件，我们可以将上图的A用户test1.txt的内容整理下，使其满足条件，然后 选择，这时test.txt.mine、test1.txt.r13、test1.text.r14将会消失。用户A就可以顺利提交了。但是，如果test1.txt是一个非纯文本文件，比如excel，这时的test1.txt将没法手动合并了，不得不放弃自己的修改。可以在test1.txt上右键选择消除掉test.txt.mine、test1.txt.r13、test1.text.r14这三个文件。（点击Resolve不会更改test1.txt以及服务器端的内容，仅仅是消除了那几个文件。）此时的test1.txt文件是可以提交的，它对应的是服务器的最新版本，即14版本（因为这是svn将服务器最新版本14和A用户修改内容合并后的结果）。但这是svn帮我们合并的，是不合法的文件。我们可以右键然后选择，然后test1.txt就会变成14版本，A用户的修改没有了，A、B、服务器的test1.txt都成为了14版本。如下图：

接下来A用户就可以再进行修改提交了。

**总结**

对于纯文本文件因版本过时提交失败的情况，我们可以选择更新一下，然后打开”自己的修改和服务器最新版合并“后的文件（如上文发生冲突时的test1.txt文件），进行手动合并，处理好后选择resolve然后提交。

对于非纯文本文件因版本过时提交失败时，我们只能牺牲一下自己，选择，然后更新到服务器最新版本，再修改提交

例如，如果sally修改了一个文件sandwich.txt,而harry也刚刚修改了这个文件的相同位置并提交到服务器。那么sally在做这个文件的update操作的时候会得到三个额外的文件sandwich.txt.mine、sandwich.txt.r1、sandwich.txt.r2。并且在提交的时候会遭到服务器的拒绝，因为这个文件的冲突问题还没有得到解决。要解决这个冲突，可以选择：  
a.手工合并SVN冲突文件（检查和修改文件中的冲突标志）。  
b.用一个临时文件（三个中的一个）覆盖你的工作文件。  
c.运行svnrevert<filename>来放弃所有的修改。  
一旦解决了你的冲突，需要通过命令svnresolved让subversion知道并删除三个临时文件。这时才可以提交。  
下面再说说手工合并SVN冲突。开始的时候让人觉得害怕，但做一段时间之后，就觉得不那么烦人了。  
**看看如下文本：**  
Mayonnaise  
Lettuce  
Tomato  
Provolone  
<<<<<<<.mine  
Salami  
Mortadella  
Prosciutto  
=======  
Sauerkraut  
GrilledChicken  
>>>>>>>.r2  
CreoleMustard一连串的大于、小于、等于号是SVN冲突标记，这些数据得全部删除才可以提交。其中,  
<<<<<<<.mine  
Salami  
Mortadella  
Prosciutto  
=======是你在冲突区里面做的修改。  
Sauerkraut  
GrilledChicken  
>>>>>>>.r2  
是别人在冲突区做的修改。  
在SVN冲突区中，或许你需要和你的同事沟通来安排冲突区的文本内容，如果是程序代码，你需要和同事商量一下，中间的这段代码到底应该是什么样子的。  
所有冲突区得到合理的解决之后，你就可以提交你的文件了。

**版本冲突原因：**

假设A、B两个用户都在版本号为100的时候，更新了kingtuns.txt这个文件，A用户在修改完成之后提交kingtuns.txt到服务器，这个时候提交成功，这个时候kingtuns.txt文件的版本号已经变成101了。同时B用户在版本号为100的kingtuns.txt文件上作修改，修改完成之后提交到服务器时，由于不是在当前最新的101版本上作的修改，所以导致提交失败。

**版本冲突现象：**

冲突发生时，subversion会在当前工作目录中保存所有的目标文件版本[上次更新版本、当前获取的版本(即别人提交的版本)、自己更新的版本、目标文件]。

假设文件名是kingtuns.txt

对应的文件名分别是：

kingtuns.txt.r101

kingtuns.txt.r102

kingtuns.txt.mine

kingtuns.txt。同时在目标文件中标记来自不同用户的更改。

**版本冲突解决：**

场景：

1、现在A、B两个用户都更新kingtuns.txt文件到本地。

2、文档中原始文件内容如下：

3、A用户修改文件，添加内容“A用户修改内容”完成后提交到服务器

4、B用户修改文件，添加内容“B用户修改内容”完成后提交到服务器

B用户提交更新至服务器时提示如下：

B用户将文件提交至服务器时，提示版本过期：首先应该从版本库更新版本，然后去解决冲突，冲突解决后要执行svn resolved（解决），然后在签入到版本库。在冲突解决之后，需要使用svn resolved（解决）来告诉subversion冲突解决，这样才能提交更新。

解决冲突有三种选择：

A、放弃自己的更新，使用svn revert（回滚），然后提交。在这种方式下不需要使用svn resolved（解决）

B、放弃自己的更新，使用别人的更新。使用最新获取的版本覆盖目标文件，执行resolved filename并提交(选择文件—右键—解决)。

C、手动解决：冲突发生时，通过和其他用户沟通之后，手动更新目标文件。然后执行resolved filename来解除冲突，最后提交。

解决步骤如下：

1、  在当前目录下执行“update”（更新）操作

2、  在冲突的文件上（选中文件--右键菜单—TortoiseSVN—Edit conflicts（解决冲突）），出现如下窗口

Theirs窗口为服务器上当前最新版本

Mine窗口为本地修改后的版本

Merged窗口为合并后的文件内容显示

3、  如果要使用服务器版本，在Theirs窗口选中差异内容，右键，选择Use this text block（使用这段文本块）。

同理如果要使用本地版本，在协商后，在Mine窗口右键，选择Use this text block（使用这段文本块）。

4、  修改完成后，保存kingtuns.txt文件内容。

5、  在B用户的冲突目录下，选中文件--右键菜单—TortoiseSVN—Resolved（解决）。会列出冲突的文件列表，如果确认已经解决，点OK。

6、  冲突解决

7、提交解决冲突后的文件。

如何降低冲突解决的复杂度：

1、当文档编辑完成后，尽快提交，频繁的提交/更新可以降低在冲突发生的概率，以及发生时解决冲突的复杂度。

2、在提交时，写上明确的message，方便以后查找用户更新的原因，毕竟随着时间的推移，对当初更新的原因有可能会遗忘

3、养成良好的使用习惯，使用SVN时每次都是先提交，后更新。每天早上打开后，首先要从版本库获取最新版本。每天下班前必须将已经编辑过的文档都提交到版本库。