Tortoisesvn基础使用

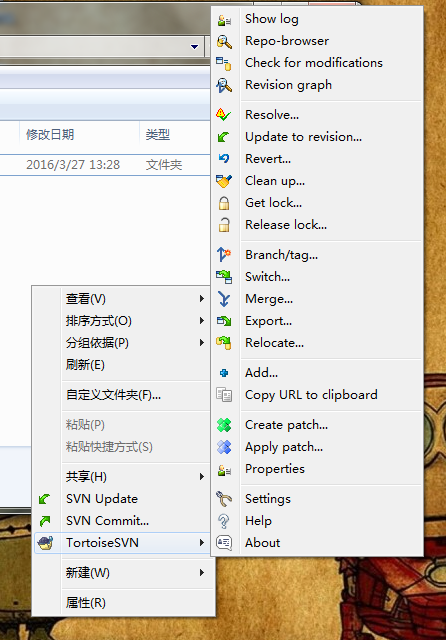
一切从版本库链接开始

首先我们会申请svn权限，然后会得到相应项目的svn检出路径

然后，在本地新建一个文件夹，在该文件夹中进行检出操作

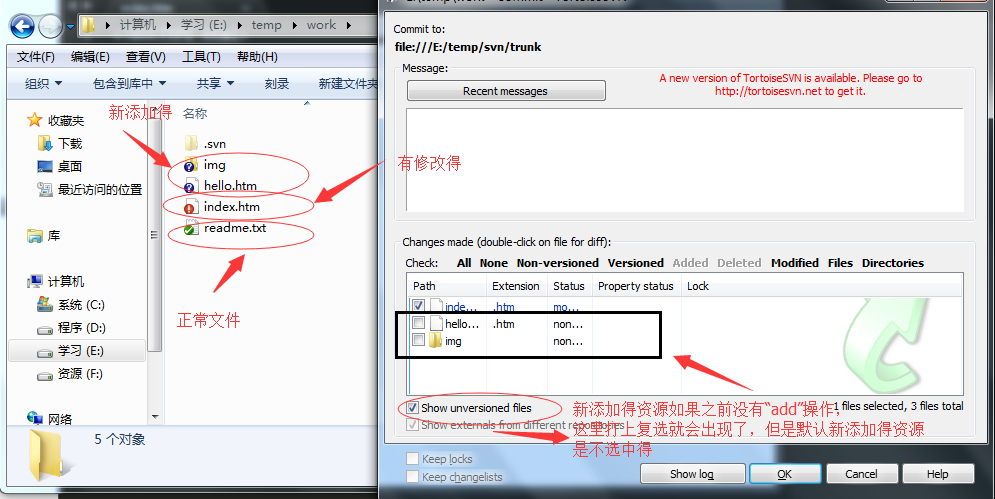
执行完这个动作，就在本地生成了一份工作副本

这时候右键点击你的文件夹就会出现svn得操作菜单



那么svn得主要操作就在这里了，接下来我按照工作的一般流程介绍下相关的功能

首先，你会修改文件，或是添加新得文件，要注意得是，新建，删除，修改，文件或是文件夹，svn得所有操作都必须提交，这些作只有在提交后，才会在svn中有所体现。新添加得文件可以用“add”操作，当然，不用这样也是可以的，在“submmit”得时候也是可以提交新的资源的，不过默认是不选中得。

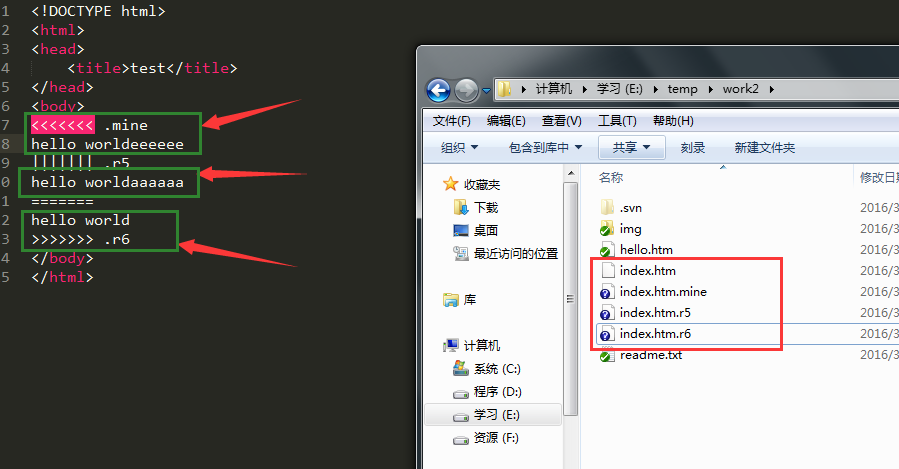


你在本地工作副本得所有操作，只会影响到你自己的工作副本，要想让大家都get到你的修改，就要将你的修改提交到svn版本库。

提交的时候有可能会遇到冲突

文件冲突

当多个人编辑了同个文件得相同行得代码，就会出现文件冲突，这时候svn并不知道要怎么处理这样的问题，所以它只是简单的把旧版本，新版本和你的修改糅合在一起，把冲突留给你处理



当你重新编辑文件后，可以执行下“resolved”表示已经处理了当前冲突，然后就会发现\*\*.mine

\*\*.r\*

这样的文件就主动删除了，接着再提交一遍就可以了。

如果你想完全避免这样的冲突，就可以用“get lock”来获取该文件得锁，这样别人要想编辑这个文件得时候就会提示不能操作，用锁得弊端就是不能并行开发，只能等你修改完了，释放锁了，另一个人才能接着工作，所以并不推荐，相比于文件冲突成本还是较高得。

树冲突

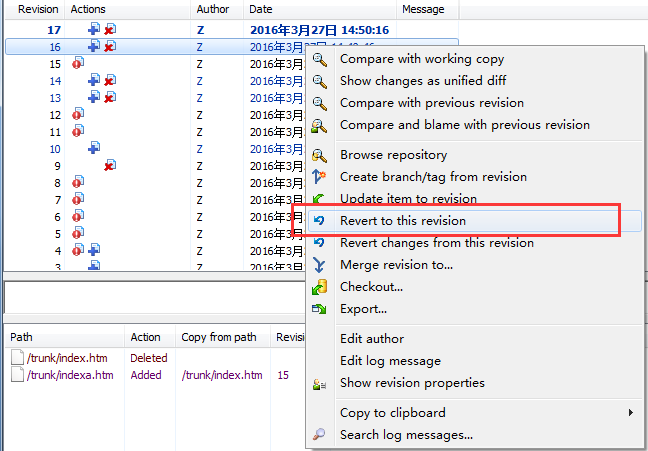
树冲突，相对于上面提到得文件冲突，就是目录结构得改变引起得冲突，主要就是文件或文件夹得重命名或删除引起得。

当你提交的时候，svn会比较你的工作副本和版本库得版本号，发现你的版本号不是最新得就会提醒你进行更新动作，这时候如果发现你的目录结构和版本库的目录结构不一致就会报树冲突。

撤销修改

如果你对自己的修改不满意，并且还没提交到版本库，那么可以通过“revert”进行撤销

如果已经提交到版本库，那么就要到日志中找到相应的版本



执行完这条命令后，本地工作副本就会返回到指定的版本

然后再提交，覆盖上一个版本。

Svn得分支

为什么要用分支？

当开发新功能得时候，就需要有分支，如果全部在主干上修改，这时候测试人员就会有疑问，这是bug还是你在修改，耽误人家工作。而这时候发布人员恰巧发布了，那就悲剧了

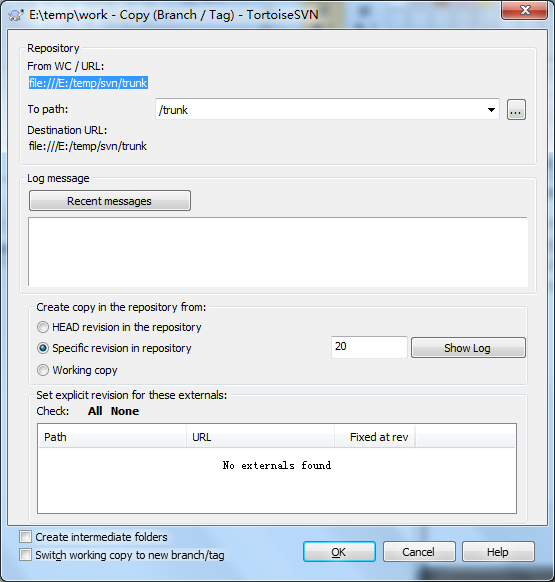
所以一般我们会有这样几个分支

一个是主干分支，这个分支主要用作开发，调试

一个是测试分支，在主干分支上调试完成后将代码上传到测试分支，用于测试人员测试

一个是发布分支，测试人员觉得功能ok了，就可以讲测试分支合并到发布分支了

可以用“branch/tag”新建分支



Svn分支合并

当你在分支上的工作完成后，就可以将分支上的修改合并到主干上来了

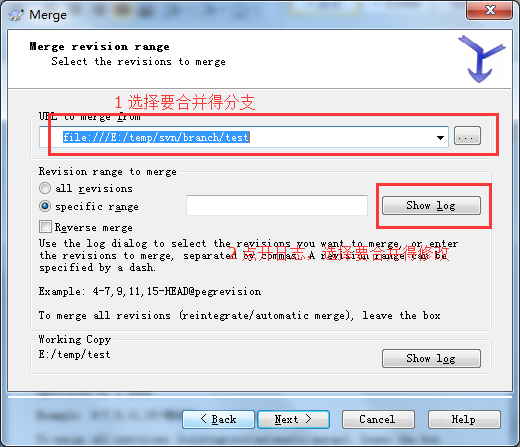
这里要说下svn得“switch”功能，这个功能就是再各个分支间进行切换

但是，因为svn得切换并不方便，同时也不能很方便得知道当前目录所属得分支，

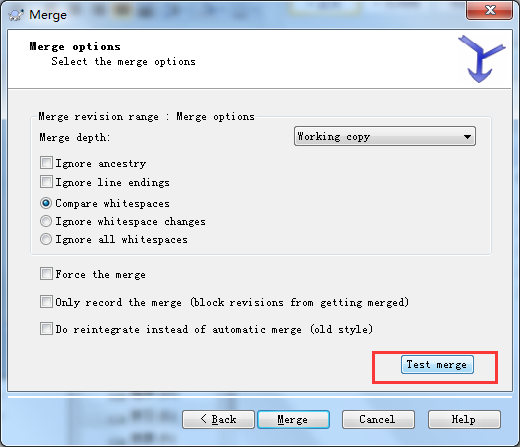
所以这个功能我们并不常用，或者说我们几乎不用

所以，一般有几个分支，我们就检出几个目录

那么，要想将分支A得修改合并到分支B，那么先进入到分支B得目录，选择“merge”命令,选择“merge a range of revisions”



合并过程也可能出现冲突，不要方，按照上面介绍得方法进行修改就ok了。



合并之前可以点击“test merge”预览下合并得效果，如果没问题，就可以正式合并了