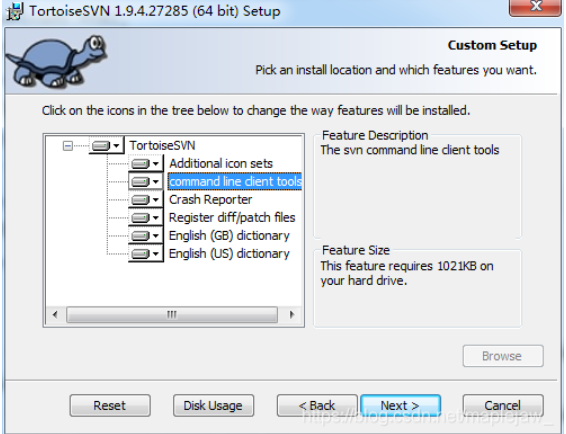
## 安装说明

首先安装SVN客户端，windows一般选择乌龟客户端https://tortoisesvn.net/downloads.html。

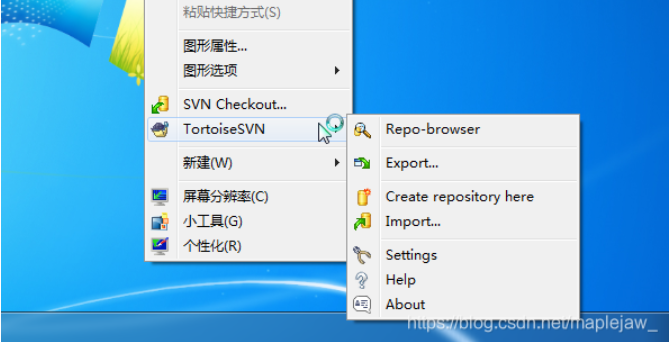
根据系统位数选择相应客户端进行安装。



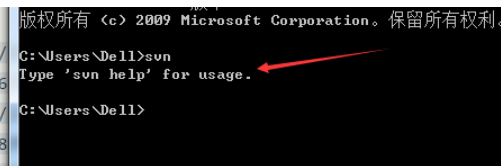
如果你喜欢用命令行操作，请务必记得勾选command line client tool为will be install on local hard driver,不用命令行的跳过这一步。



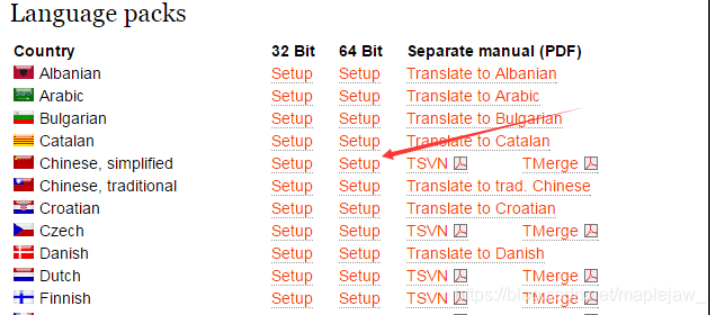
然后一路next即可安装。安装完毕后，在任意地方右键查看快捷菜单。发现TortoiseSVN即表示安装成功。



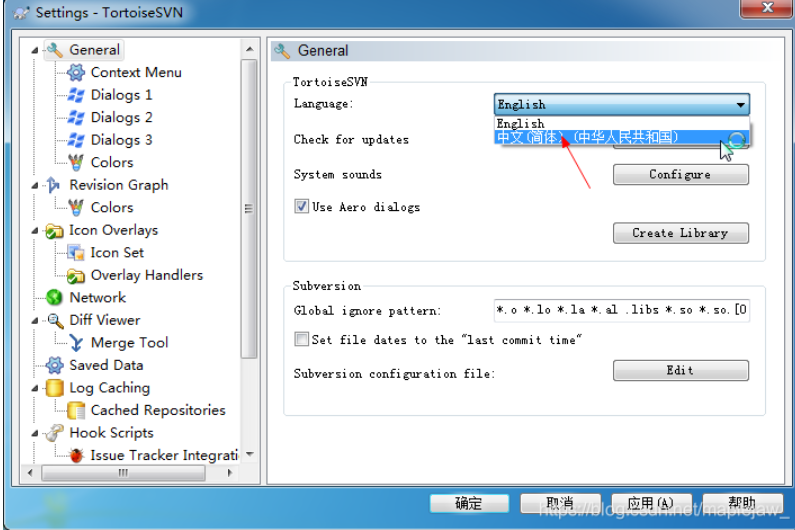
如果勾选了安装命令行工具，那么输入命令SVN，有如下提示也表示安装成功



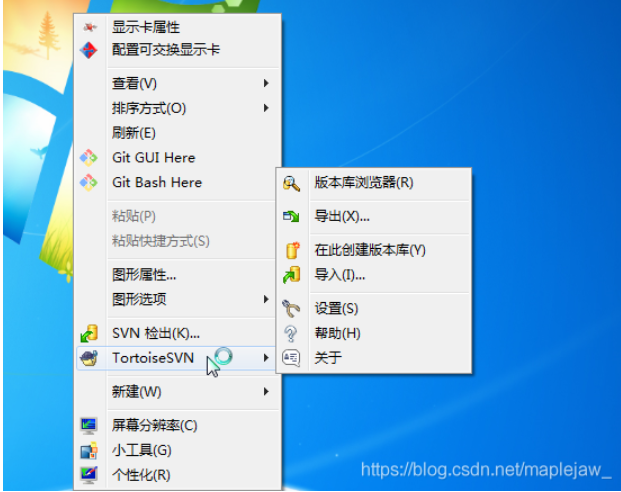
但是此时菜单全是英文的，如果你不习惯英文，可以去下载语言包，记得下对系统位数。



安装完语言包之后，可以右键进入setting设置。



选择你喜欢的语言（比如中文），然后确定，不出意外，现在的语言已经切换到中文了。



安装教程到此结束，下面开始介绍SVN的使用。

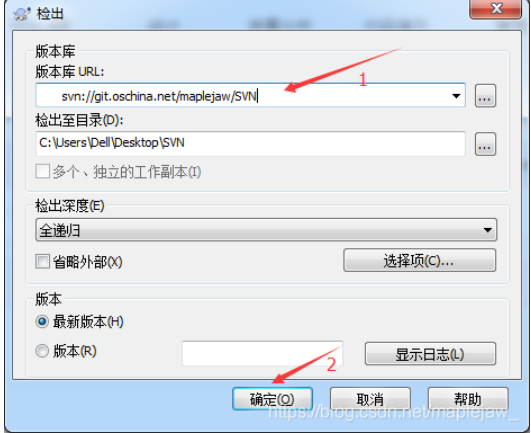
**使用说明**

**检出项目**

假如项目已经在服务器的仓库里，那么现在你要做的就是把它检出到本地。  
首先创建一个空文件夹。在空文件夹内右键，选择SVN检出。



现在你看到应该是这个界面，填入版本库地址，选择确定。



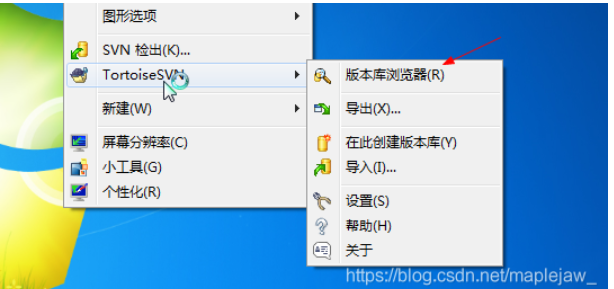
此时会弹出一个对话框让你输入账号密码，输入你的账号密码即可。记得勾选保存认证，不然每次操作都会让你输入。



等几分钟就可以检出完毕。

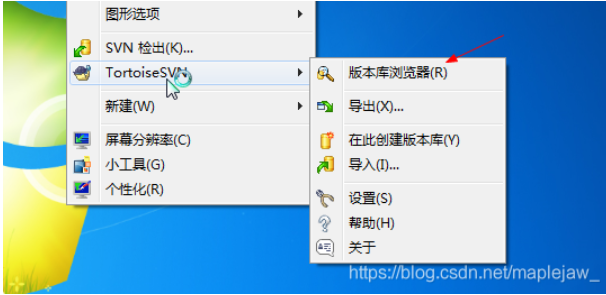


此时在你的目录下就能看到你的项目，现在可以开始愉快的工作了。

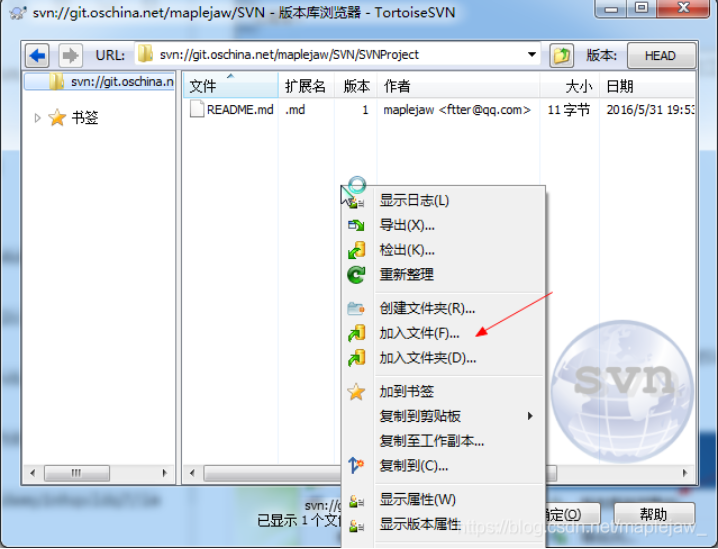


### 导入项目

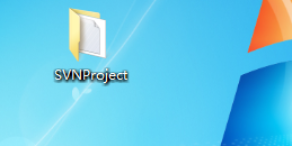
但是有时候你已经在本地建立好了项目，需要把你项目推到SVN上，此时应怎么做呢？  
右键选择版本库浏览器。



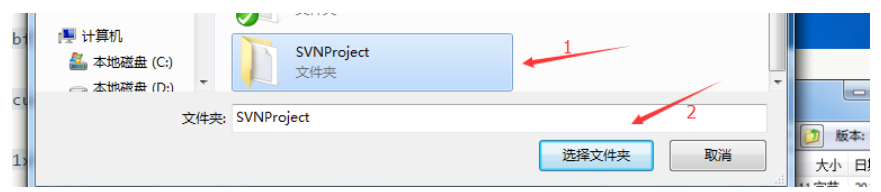
在相应目录下，右键，加入文件/加入文件夹，选择相应目录即可。



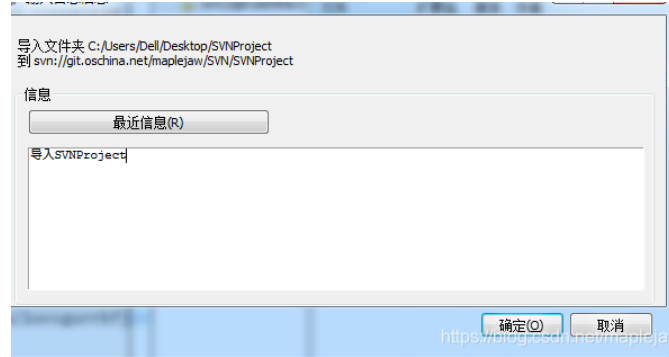
比如我现在有个项目叫SVNProject，我想把它传到SVN上。



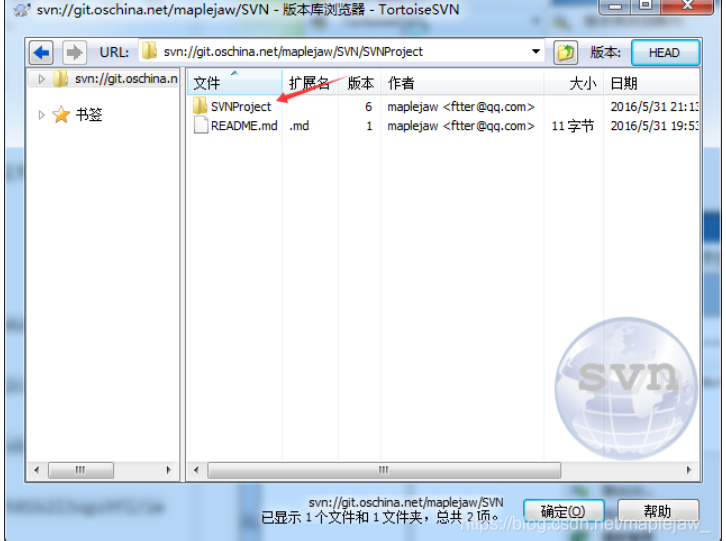
那么我只需选择加入文件夹即可。



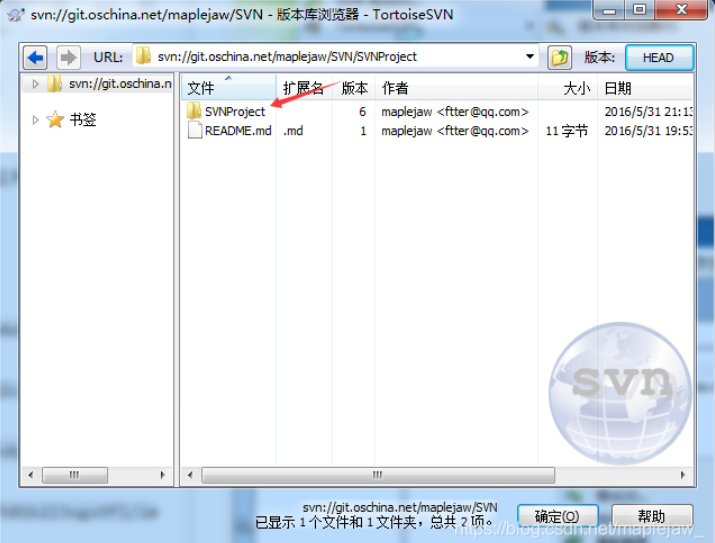
务必要输入提交信息。这样别人才能知道你干了什么。



导入成功就能看到目录。



但是，不要以为导入成功就可以了。你还得重新检出，重新检出的项目才是受SVN控制的，务必记得检出。



在SVNProject上右键检出到本地，然后在里面进行修改。现在就可以愉快的工作了。  
检出过后的右键菜单变成了这样。

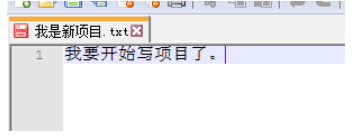


### 提交

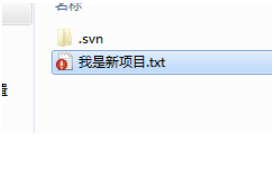
绿色表示当前文件没有被修改过（看不见颜色的重启下电脑就好了）。



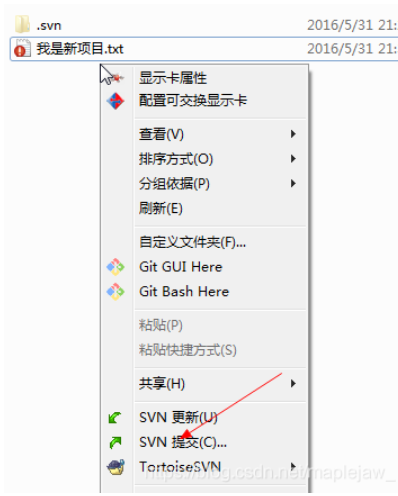
假如我现在在我是新项目.txt中加了一行字，然后保存。



发现现在变成了红色，红色表示已修改。



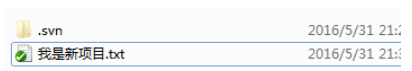
怎么提交修改？  
在根目录下，右键选择提交。



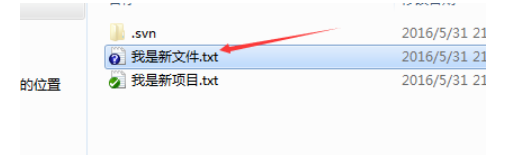
务必记得输入提交信息（虽然不输入也能提交），提交信息可以方便日后查看。



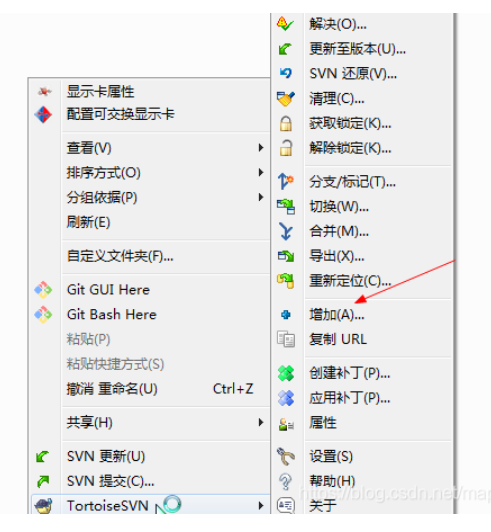
提交完毕后，可以发现又恢复到了绿色。



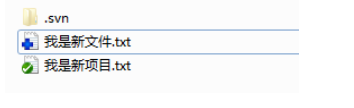
假如现在加入了一个新文件。可以看出是蓝色的。蓝色表示不属于版本库的未知文件，未知文件是不能提交的。



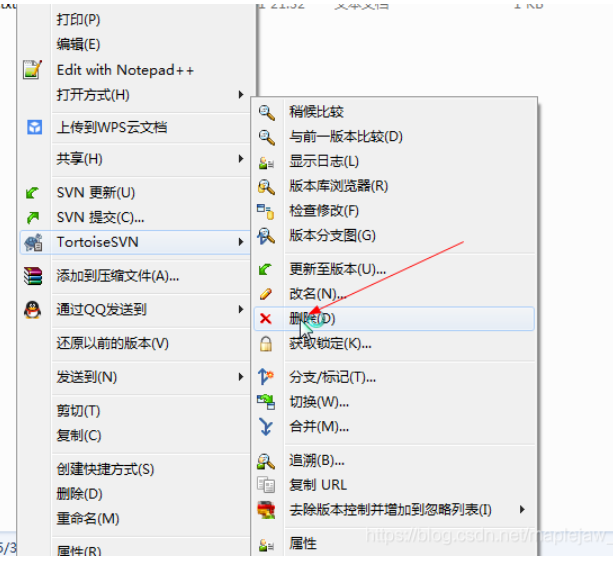
记住选择增加把它加入到版本库里面去。



增加完毕后，变成了蓝色加号，表示新增加的版本库文件。



接下来，只需写代码，然后提交即可。  
删除文件也应该右键提交，如下。

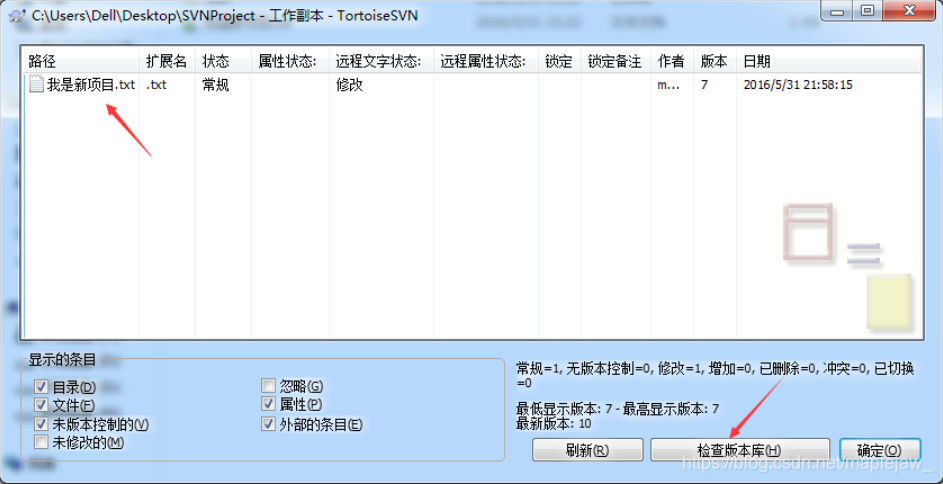


记得随时检查你的文件状态，如果没有添加到版本控制里要及时添加进去，不然你的文件提交不上去。

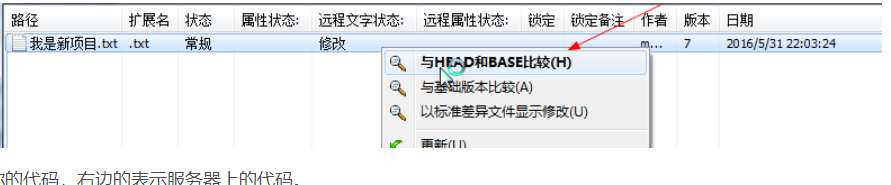
### 更新

假如你和B同学在协作。B同学写完代码提交到了SVN上，如果你想获取最新修改，就需要选择更新（如果服务器上已经有别人提交过的新的，你是提交不上去的，必须先更新再提交）。

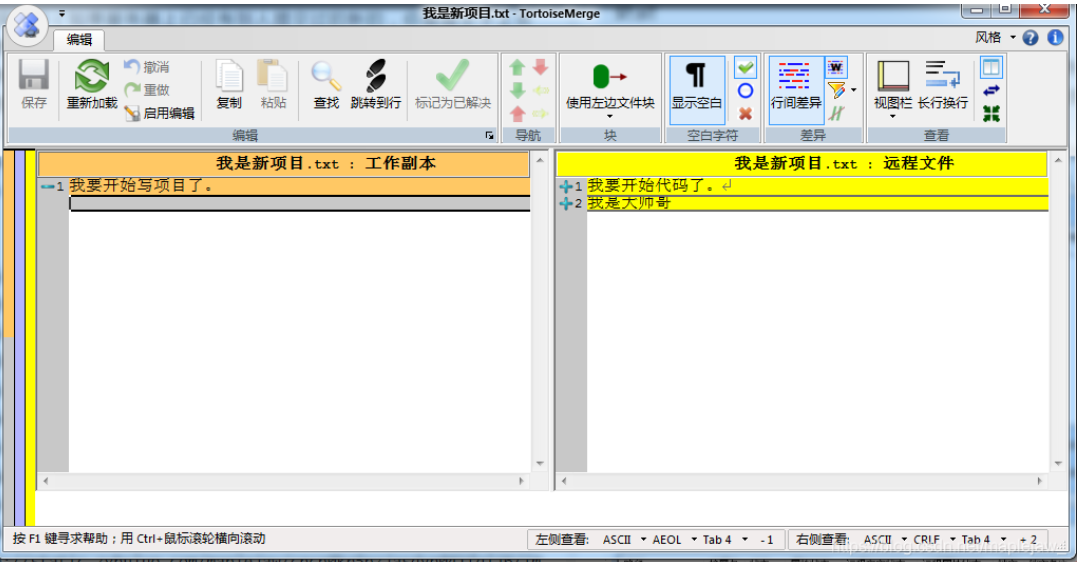
怎么知道服务器有没有更新？你可以直接选择更新，有没有更新一下就知道。或者右键检查修改，然后检查版本库，就能看到服务器上改了哪些文件。



右键选择HEAD和BASE比较。

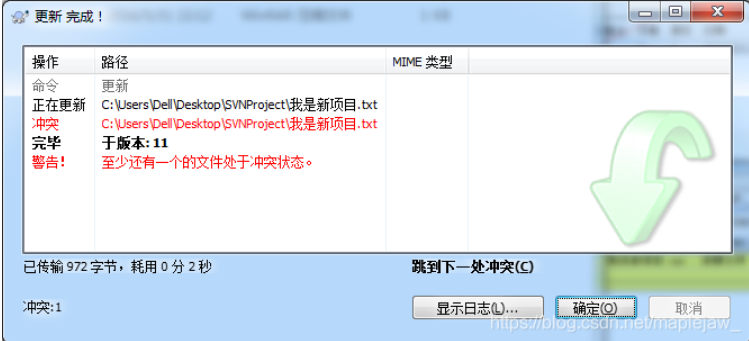


左边的表示你的代码，右边的表示服务器上的代码。

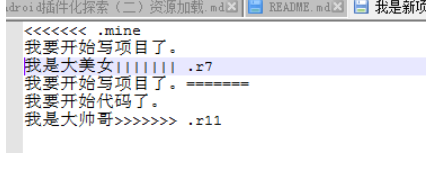


如果有修改记得及时更新到本地然后再继续工作。

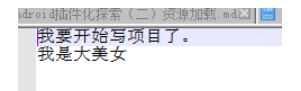
但是有时候更新会冲突，比如你和服务器上的改了同一个地方。  
这时候你需要更新下来解决冲突。



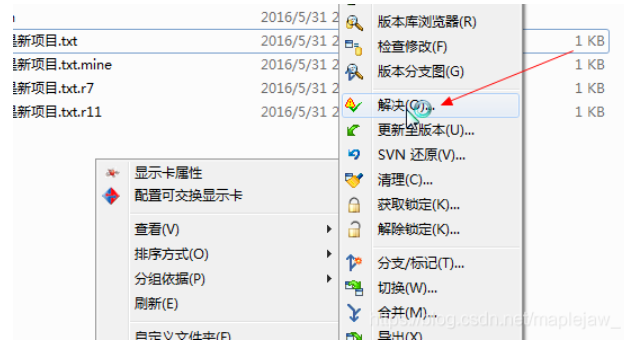
它会提示你哪个文件冲突，你只需打开那个文件，按照需求解决冲突即可。



<<<<<<.mine到====表示你的代码，其他表示服务器的代码。你只需改成你想要的。



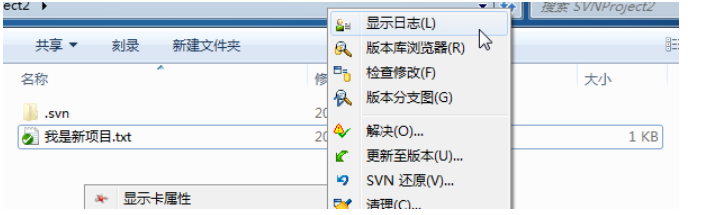
然后选择解决，告诉SVN我已经解决冲突了就行了。



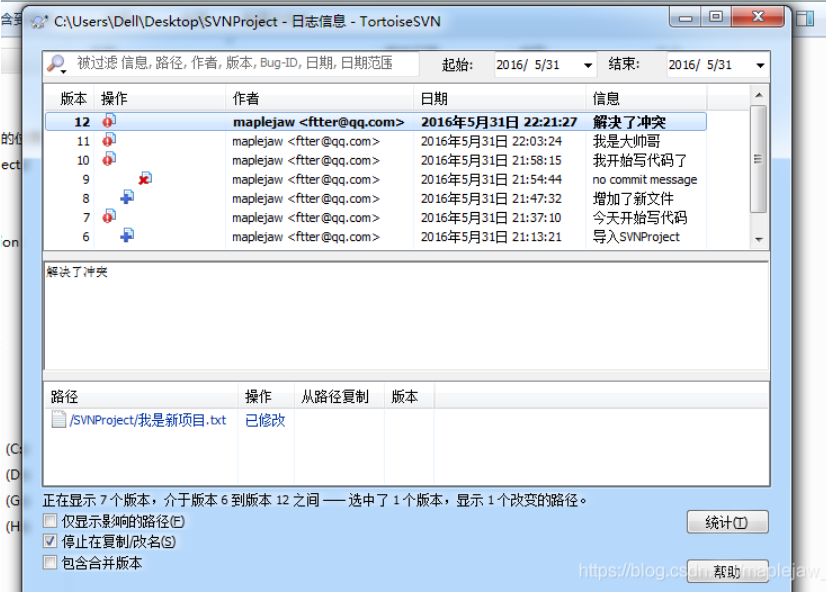
剩下的就是团队协作间的更新提交操作，这里不做赘述。

### 查看日志

选择显示日志，可以看出团队里面的人干了什么。



可以看出谁谁谁，什么时间，干了什么事。最后那一列信息是自己提交的时候写的。建议大家提交时务必要填写提交信息，这样别人一看就知道你干了什么。提交信息对于自己也是有好处的，时间长了也能看到当初做了什么。

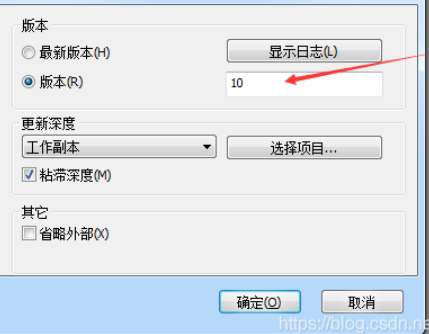


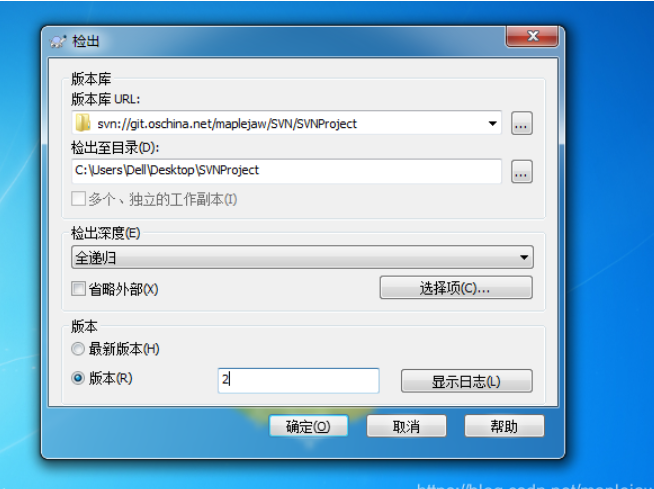
### 版本[回滚](https://so.csdn.net/so/search?q=%E5%9B%9E%E6%BB%9A&spm=1001.2101.3001.7020)

如果你改了东西，但是还没有提交，可以使用还原功能。



但是如果我们写错了东西并且提交了上去怎么办？通过版本回滚可以将文件恢复到  
以前的版本。右键更新至版本，通过查看日志来选择版本，然后回滚即可。

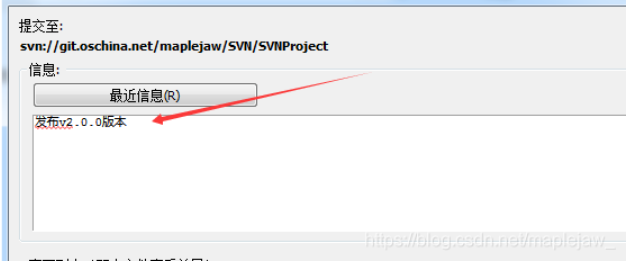


有时候我们需要查看以前版本的代码。此时我们可以新建个文件夹检出到指定版本。

### 版本控制

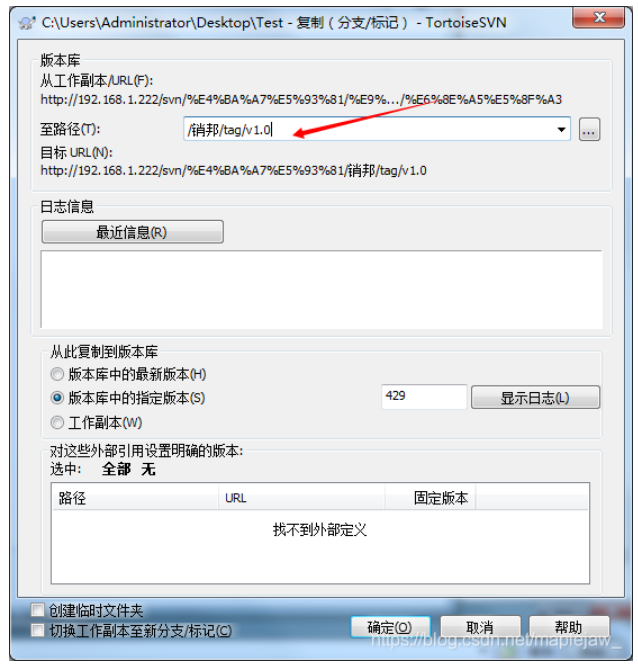
版本控制有好几种方法，如下。

1. 在提交发布版本时添加版本信息，这是最简单的一种方法。

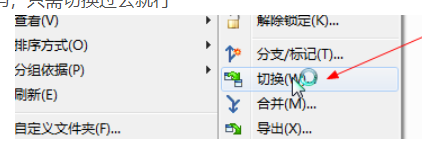


打标签  
每次发布版本时应该打标签。右键选择分支/标记。在至路径以版本号打上标签即可





这样你就有了一个v1.0版本的标签。  
以后如果你想查看某个版本的代码，只需切换过去就行



### 总结

我们在日常使用中，最常用的是更新和提交操作。这两个步骤务必要非常熟练。其他的可以在遇到问题是查看文档。此外，需要注意的是，所有版本控制工具只能跟踪文本文件（能用记事本打开查看的文件），不要妄想SVN能记录你word改了哪一行。一旦遇到word冲突，记住仔细对比两个版本，然后解决冲突。