# 1.TortoiseGit图形化客户端工具使用

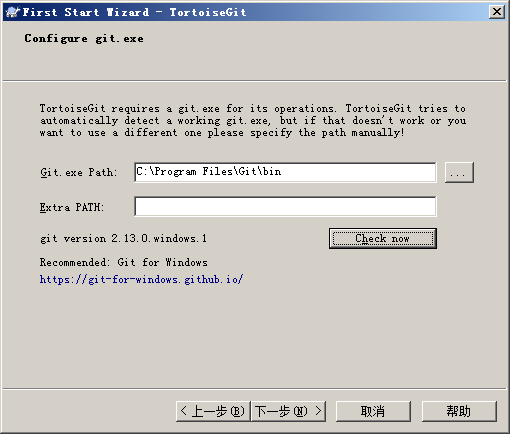
### 图形化界面Git操作

#### TortoiseGit安装

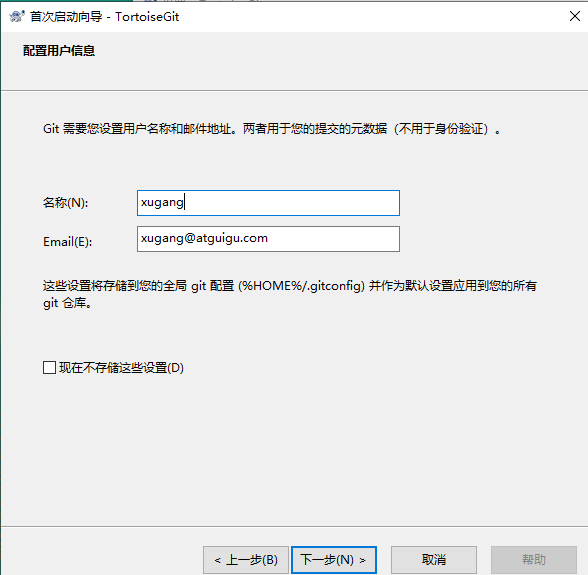
又称为小乌龟：TortoiseGit-2.4.0.2-64bit.msi

一路“Next”使用默认选项即可。

配置git.exe，在上面我们已经安装过git-for-windows了所以在此找到git.exe所在的目录。

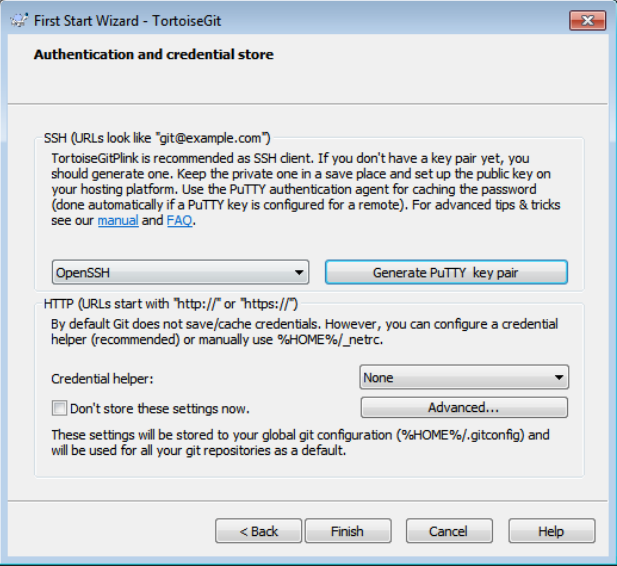


配置开发者姓名及邮箱，每次提交代码时都会把此信息包含到提交的信息中。

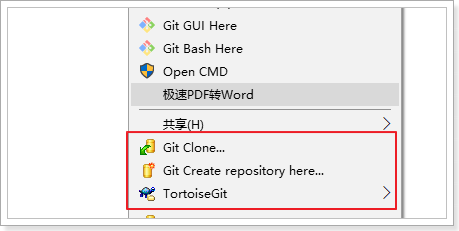


**注意：这里的用户名和邮箱和我们后面实际提交代码的用户名和密码无必然联系。**

**使用OpenSSH配置**，点击“完成”按钮完成配置。



安装完毕后在系统右键菜单中会出现git的菜单项。



**TortoiseGit安装完成后，一定要重启电脑**。

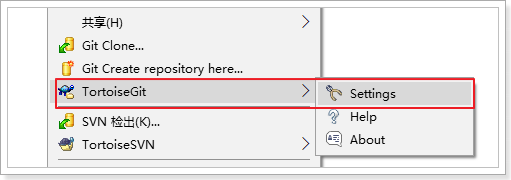
#### 安装中文语言包

安装中文语言包并不是必选项。可以根据个人情况来选择安装。TortoiseGit-LanguagePack-2.4.0.0-64bit-zh\_CN.msi

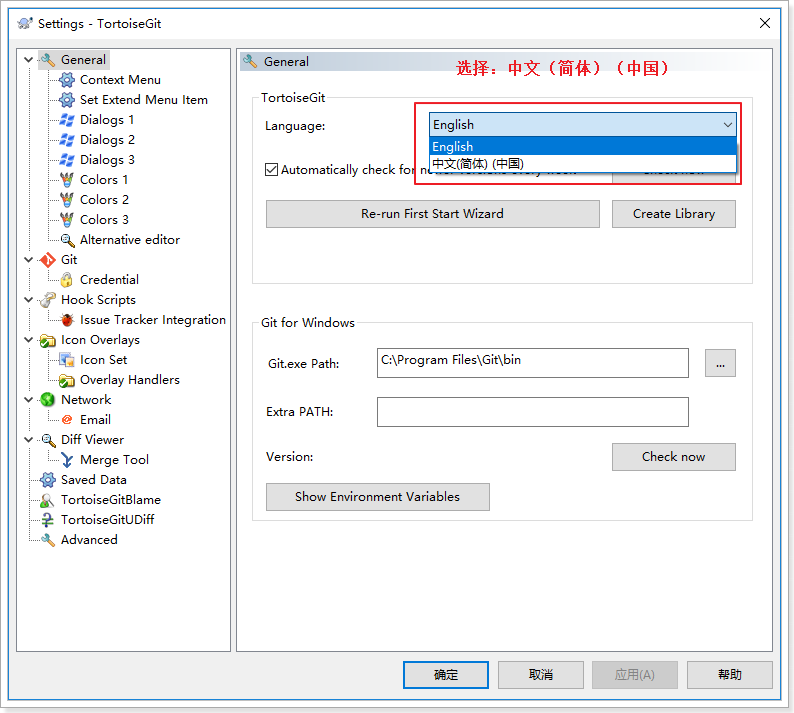


直接“下一步”安装完毕

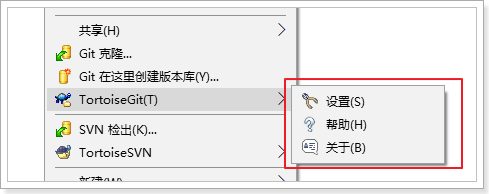
鼠标右键选择Settings



语言包安装完毕后可以在TortoiseGit的设置中调整语言

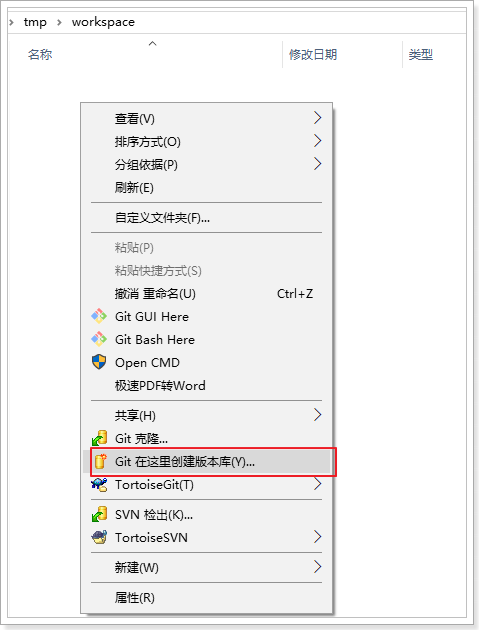


效果：



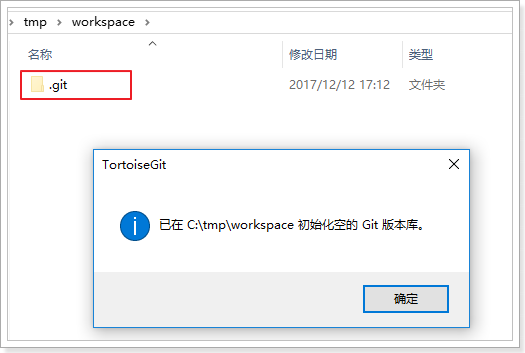
#### 创建版本库

使用TortoiseGit时只需要在目录中点击右键菜单选择“在这里创建版本库”，注意确保当期目录下没有.git目录，否则菜单中没有这一项

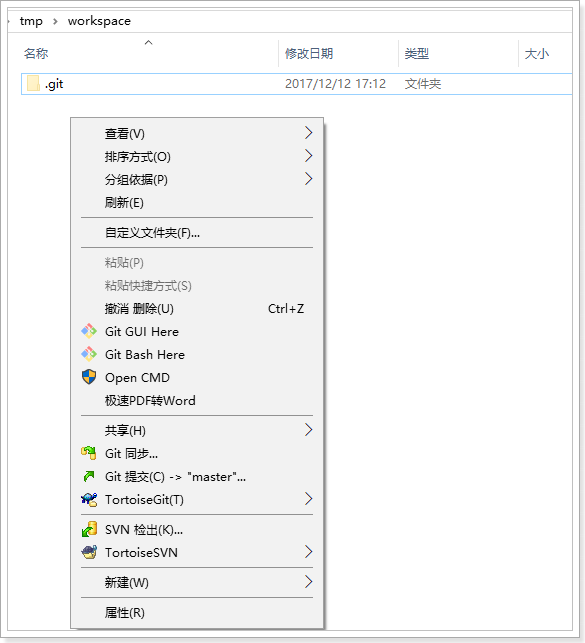




创建成功，会在此目录下创建一个.git的隐藏目录，如下：



再次右键，菜单中再也没有“git在这里创建版本库（Y）…”这一项了

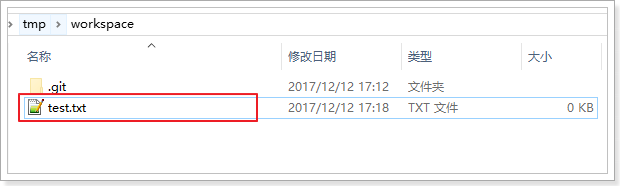


#### 查看文件状态

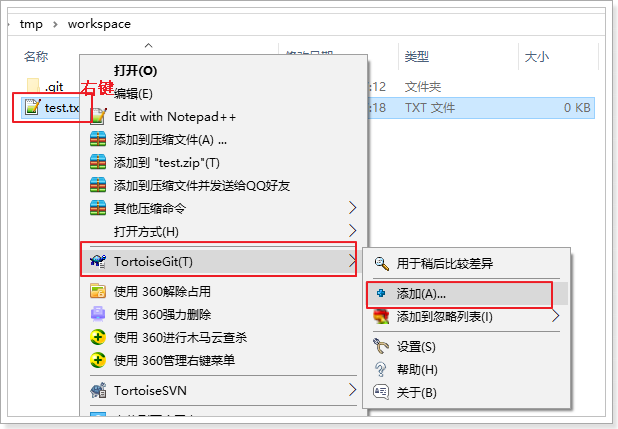
通过文件的图标变化即可查看文件状态。

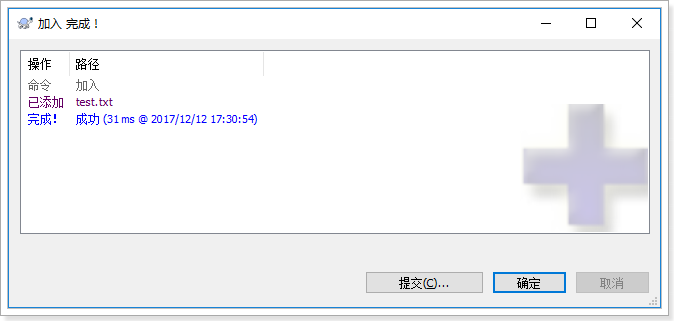
#### 将工作区文件添加到暂存区

第一步：在工作区创建文test.txt：

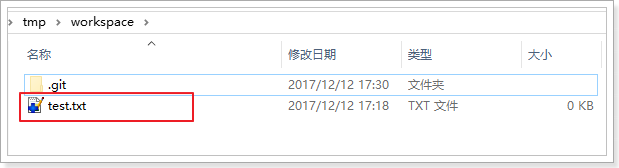


右键test.txt –> TortoiseGit –> 添加



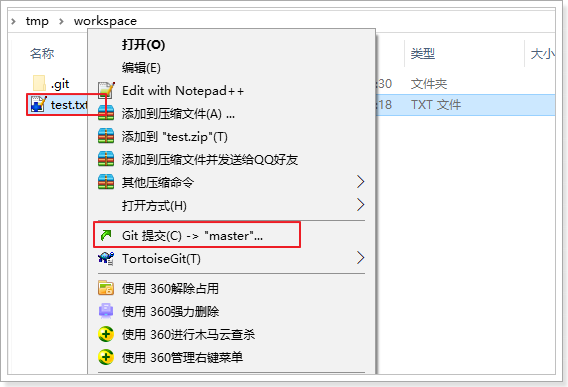


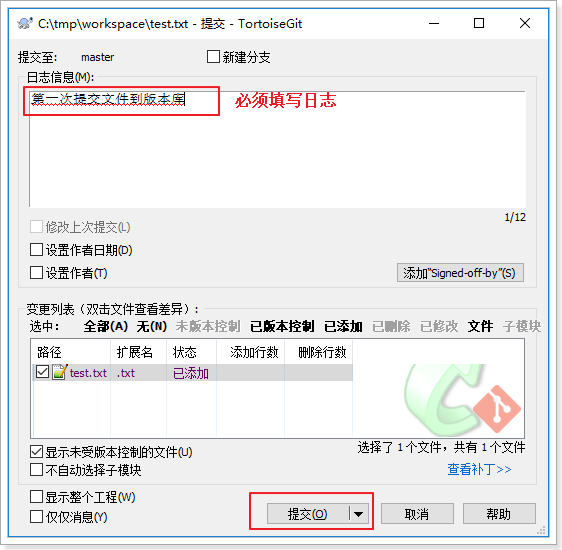
点击确定，加入到暂存区（文件变为带“+”号的图标）：

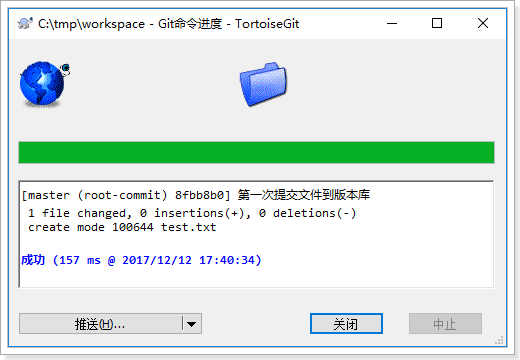


#### 将暂存区文件提交到本地库

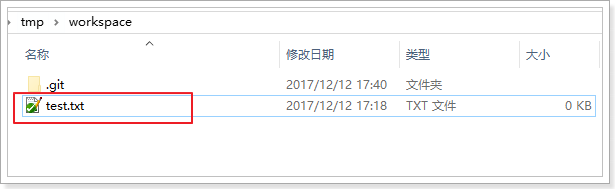
在test.txt上再次点击右键选择“提交”，此时将文件保存至版本库中。







提交后的效果：



#### 查看历史版本

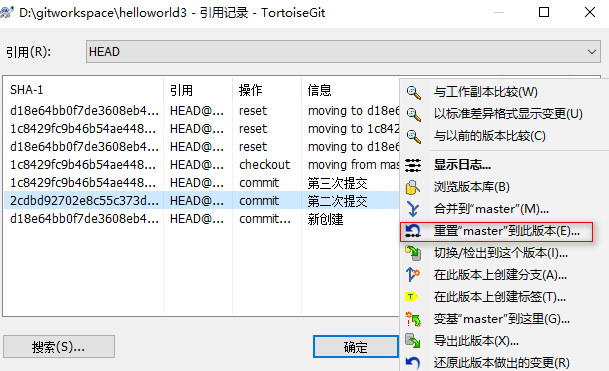
在开发过程中可能会经常查看代码的修改历史，或者叫做查看代码的修改日志。用来查看某个版本是谁修改的，什么时间修改的，修改了哪些内容。可以在文件上点击右键选择“显示日志”来查看文件的修改历史。

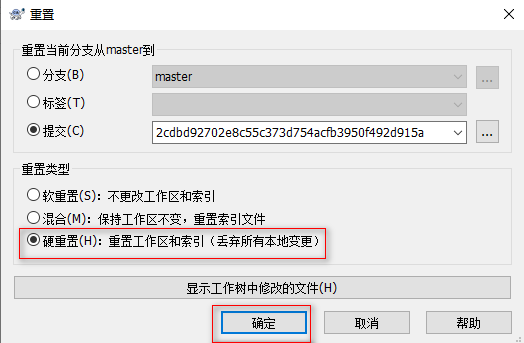




#### 回退版本

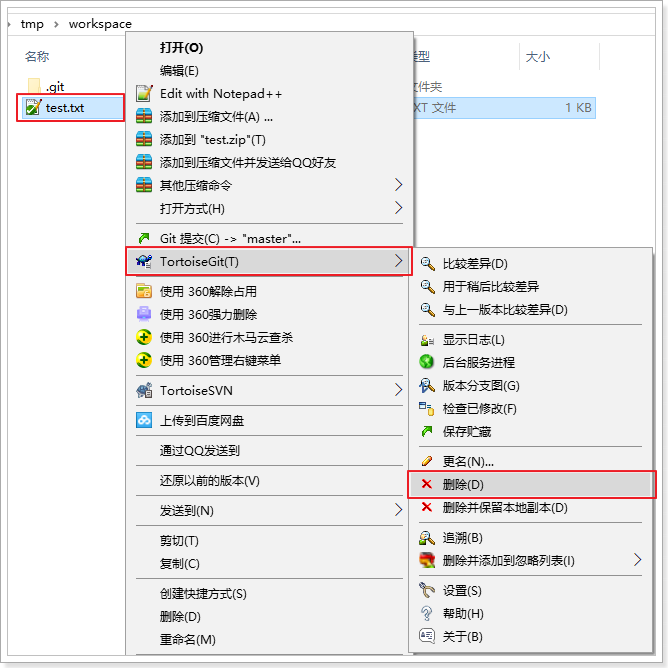




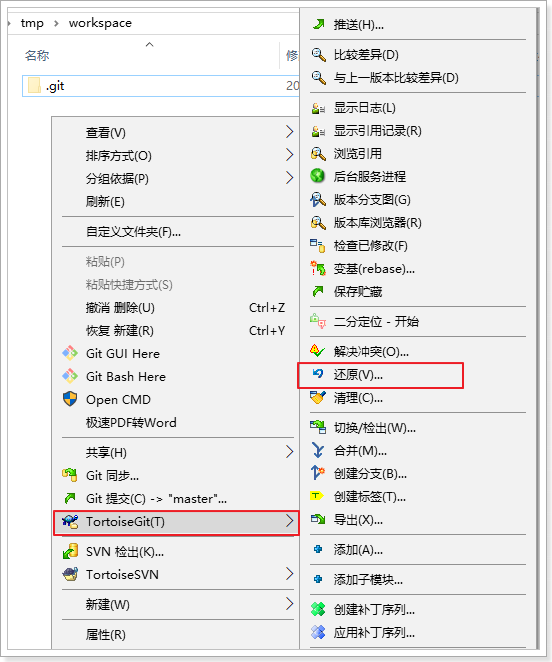


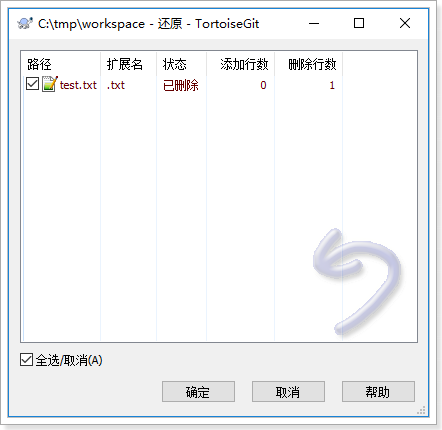
#### 删除文件

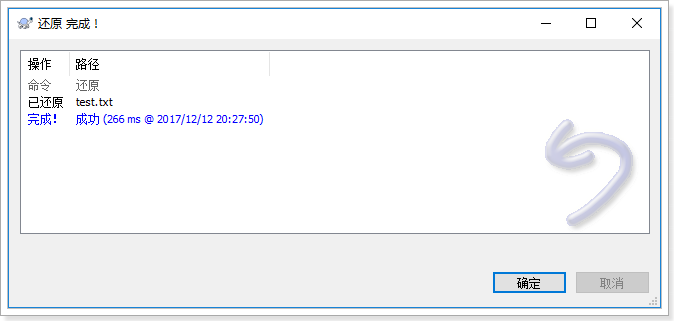
需要删除无用的文件时可以使用git提供的删除功能直接将文件从版本库中删除。



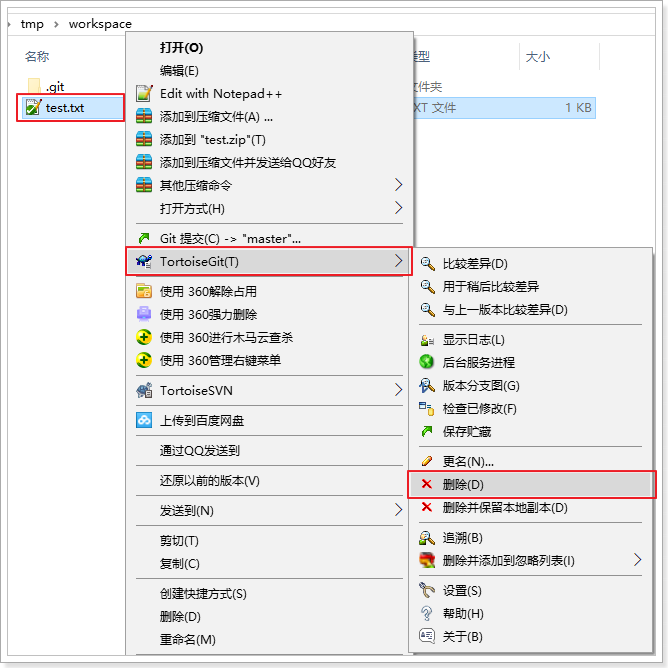
误删除，仍然可以还原：







如果要彻底删除：



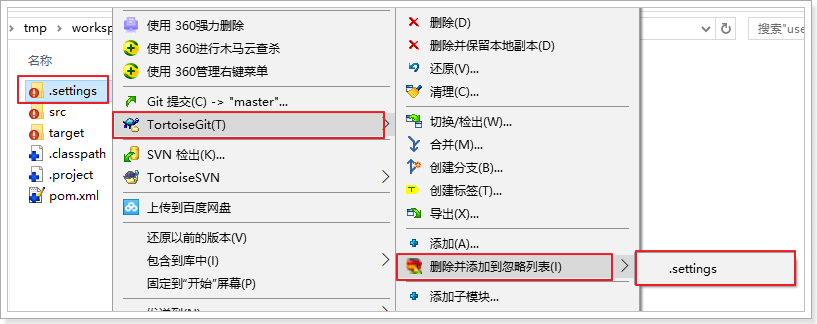


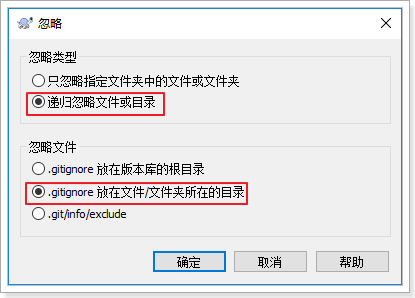


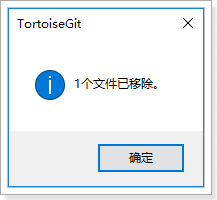
注意：即使**在这里删除了也可以使用前面的版本回退穿越回某个历史版本**。

#### 案例：将java工程提交到本地仓库中

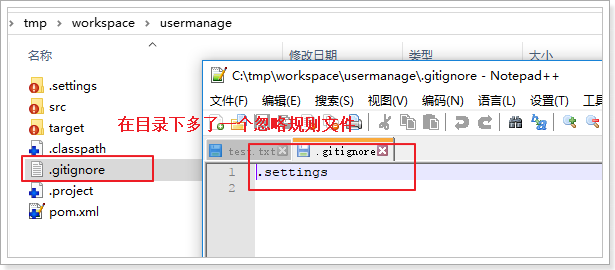
第一步：考虑要忽略的文件,创建ignore文件，记录要忽略的文件和目录







选择保留本地文件。完成后在此文件夹内会多出一个.gitignore文件，这个文件就是文件忽略文件，当然也可以手工编辑。其中的内容就是把.settings目录忽略掉。

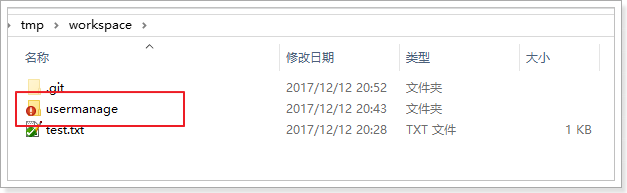


依次把target、.classpath、.project添加到忽略列表，添加完成后：

第二步：将工程添加到暂存区。

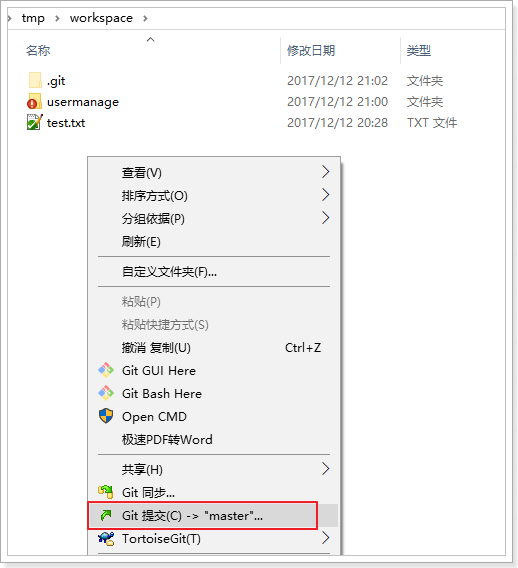


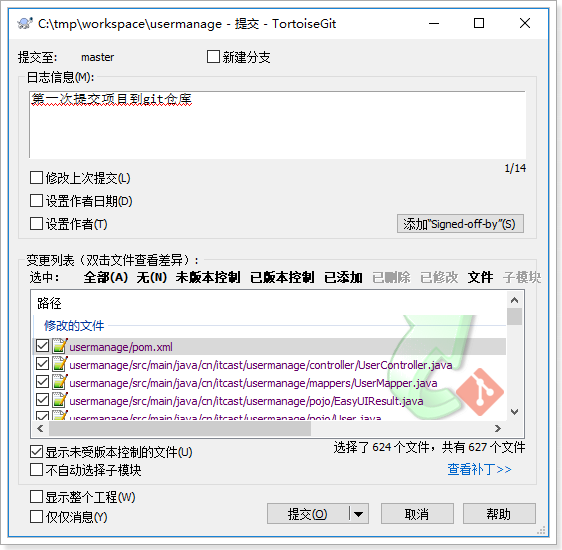
点击确定完成暂存区添加。

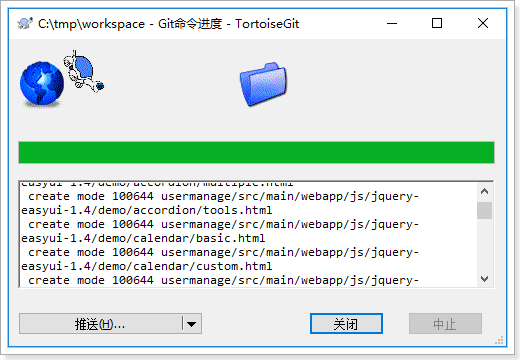


第三步：提交代码

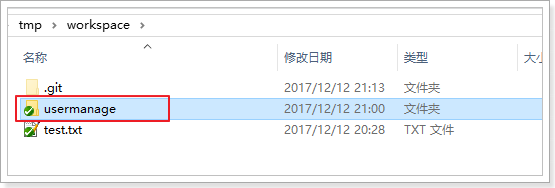
将代码添加到master分支上，其中.gitignore文件也需要添加到暂存区，然后提交到版本库。

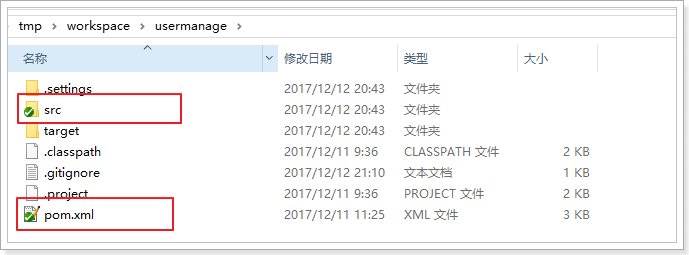






最后效果：





#### 将文件添加至忽略列表

git只是用来管理项目源码的，对于项目中的class文件或者一些其他目录都可以选择性的忽略。我们可以在工作目录中创建一个名为 .gitignore 的文件（文件名称固定），具体语法如下：

# 所有以.a 结尾的文件都会被git忽略

\*.a

# !是取反的意思，lib.a 结尾的不会被忽略 ，会被git进行管理

!lib.a

# TODO 文件会被忽略

/TODO

# build/ 目录 下面的所有文件都忽略

build/

# doc 目录下面 ，以.txt文件结尾的都会忽略

doc/\*.txt

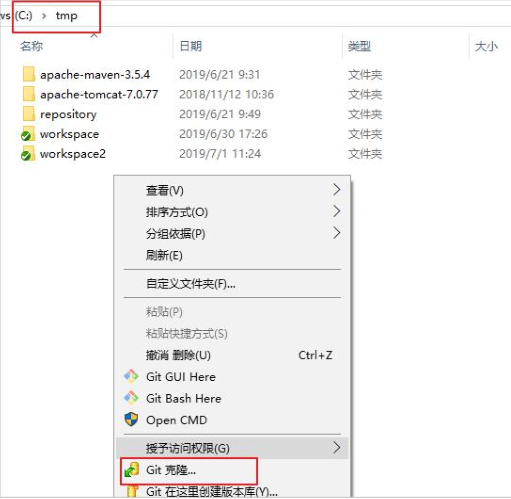
# doc 目录下面的子目录和孙子目录，以.pdf结尾的都会忽略

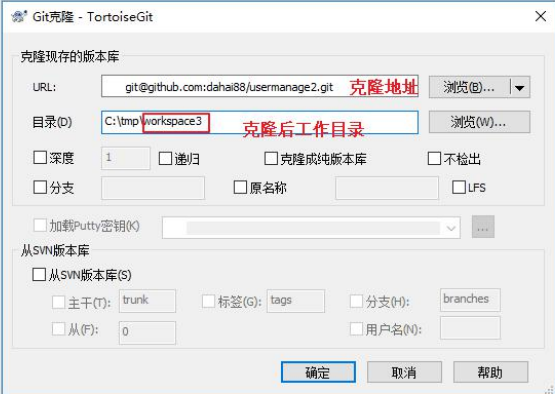
doc/\*\*/\*.pdf

### 远程仓库图形化界面操作

#### clone远程仓库

使用TortoiseGit克隆：在任意目录点击右键：





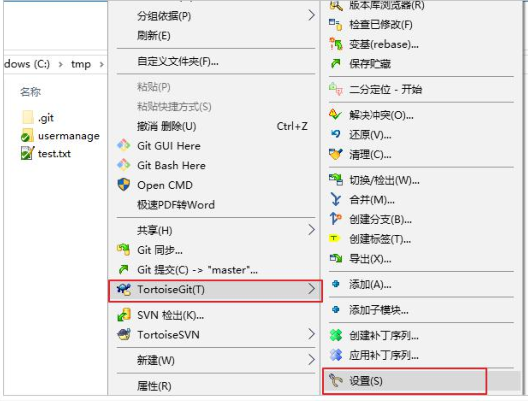
克隆效果：



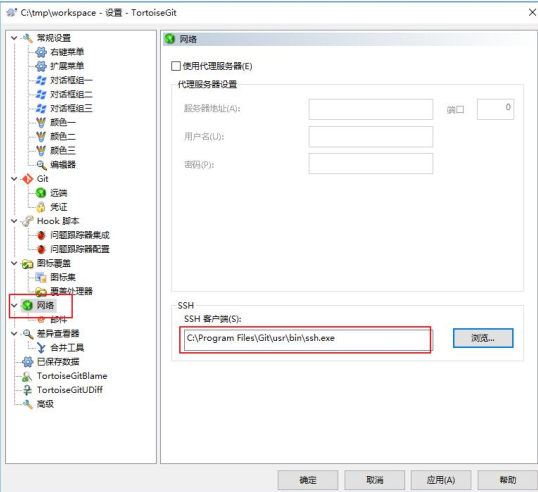
#### push到远程仓库

一、由于TortoiseGit使用的ssh工具是“PuTTY”git Bash使用的ssh工具是“openSSH”，如果想让TortoiseGit

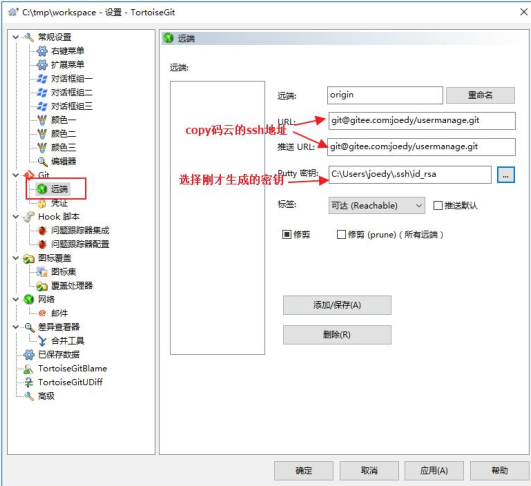
也使用刚才生成的密钥可以做如下配置：



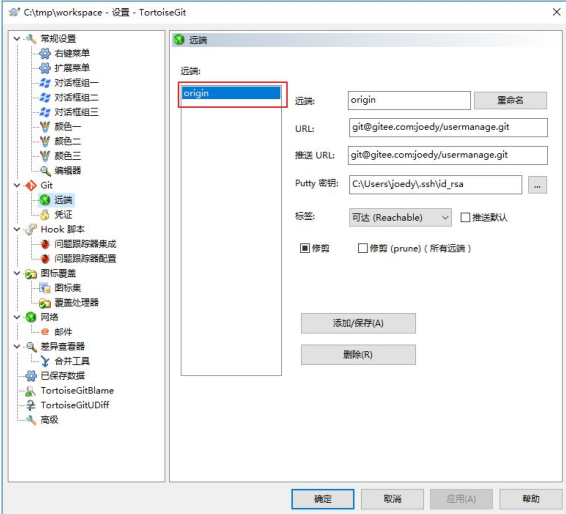
选择网路，配置 ssh 工具（git 的安装目录下）



注意：远端选项只有初始化或 clone 了仓库后才有



点击“添加/保存”按钮(注意：上面如果是基于Http协议的,我们不需要指定putty秘钥，只需要在推送的时候设置用户名和密码即可。如果使用http协议，没有提示我们输入用户名和密码，说明使用的是凭据管理器中以前保存的用户名和密码,删除以前保存和用户名和密码，然后再次推送，会提醒输入新的用户名和密码，输入即可。)

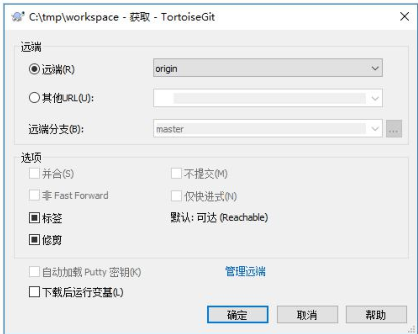


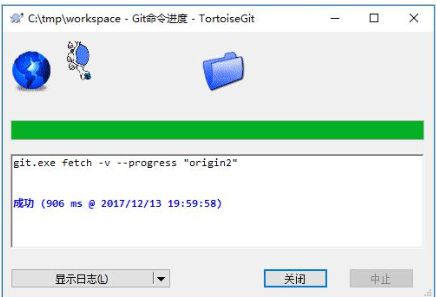
Url：远程仓库的地址

推送 URL：也是相同的

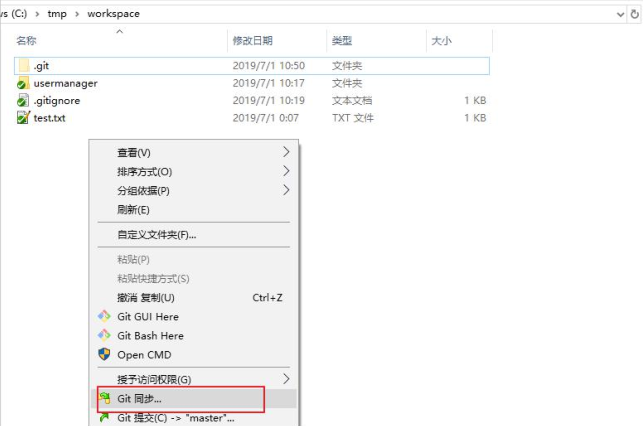
Putty 密钥：选择刚才生成的密钥中的私钥







1. 同步。在本地仓库的文件夹中单击右键，选择“Git 同步”





点击推送成功后，刷新码云的项目页面：



#### pull更新操作

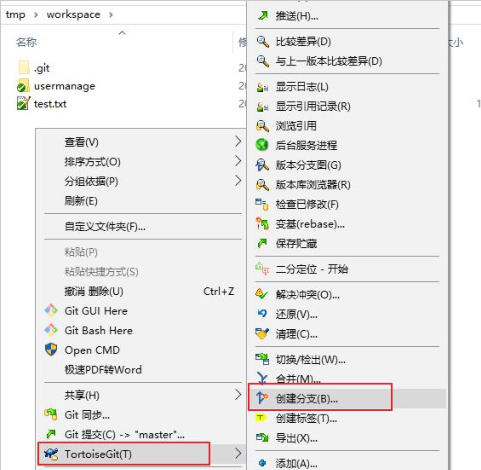
如果使用 TortoiseGit 的话可以从右键菜单中点击“拉取”（pull）或者“获取”（fetch）

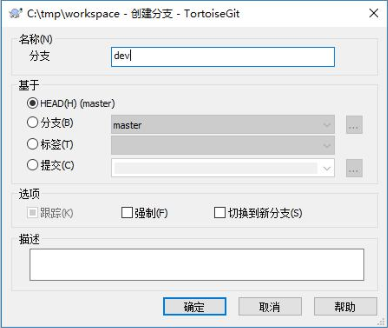


### 分支之图形化界面操作

#### 创建分支

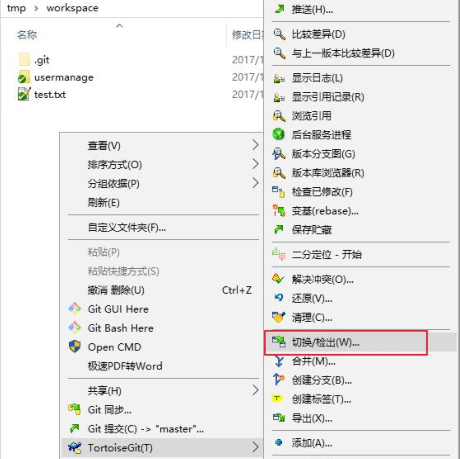
在本地仓库文件夹中点击右键，然后从菜单中选择“创建分支”：

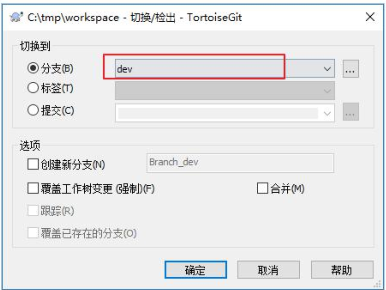




如果想创建完毕后直接切换到新分支可以勾选“切换到新分支”选项或者从菜单中选择“切换/检出”来切

换分支：

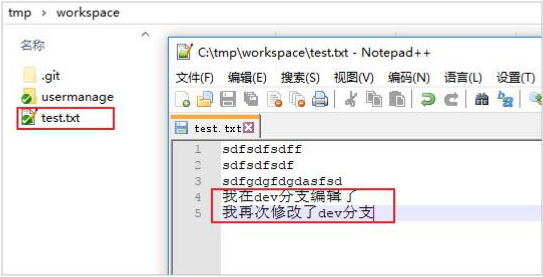




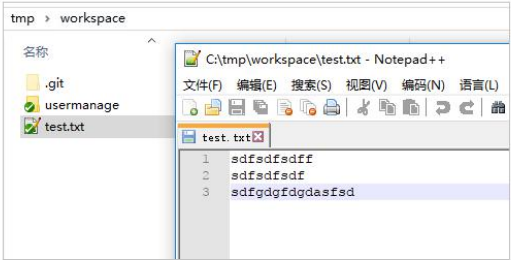
#### 合并分支

分支切换到 dev 后就可以对工作区的文件进行修改，然后提交到 dev 分支，原来的 master 分支不受

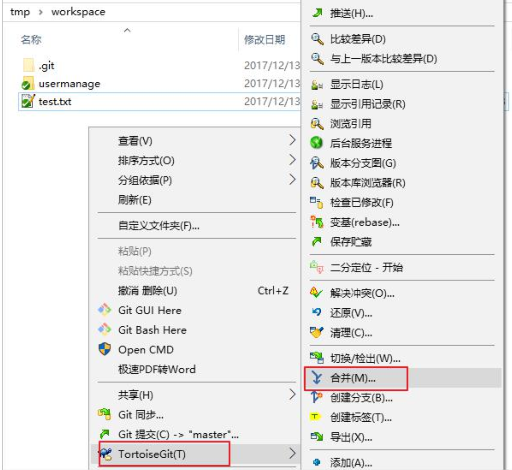
影响。例如我们修改 test.txt 中的内容，然后提交到 dev 分支。

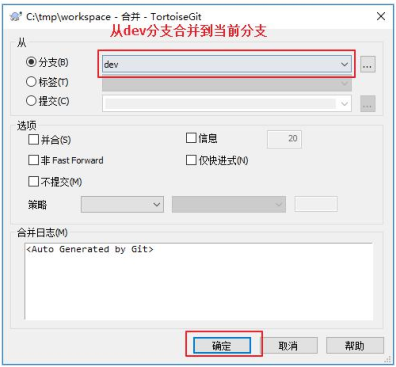


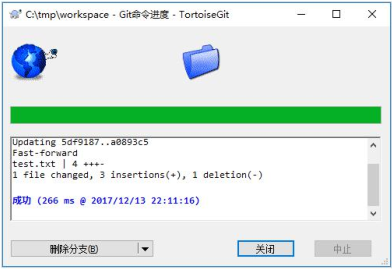
切换到 master 分支后还是原来的内容：



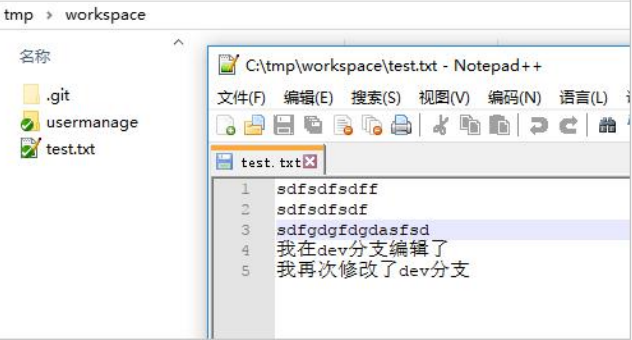
将 dev 分支合并到 master 分支，当前分支为 master。从右键菜单中选择合并:







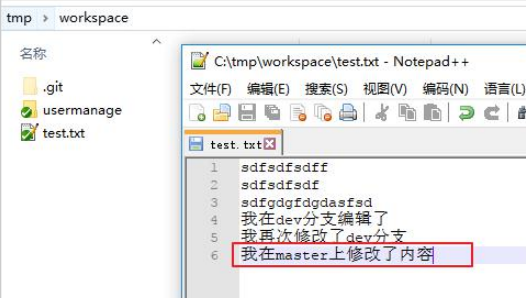
再查看 test.txt 的内容就已经更新了：



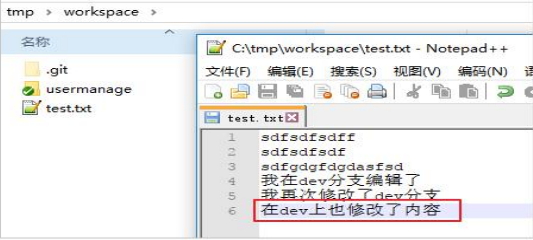
### 图形化界面之冲突解决

两个分支中编辑的内容都是相互独立互不干扰的，那么如果在两个分支中都对同一个文件进行编辑，然后

再合并，就有可能会出现冲突。例如在 master 分支中对 test.txt 进行编辑：

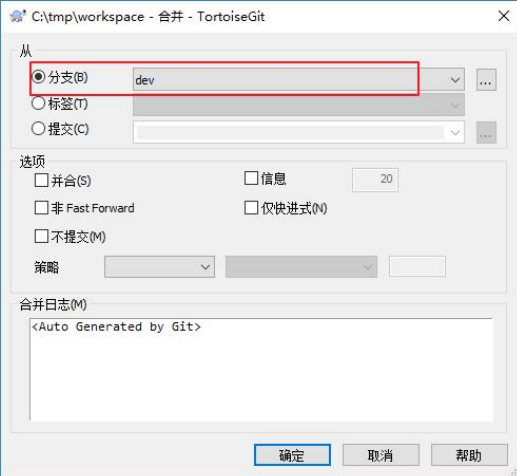


然后提交到版本库。切换到 dev 分支，对 test.txt 进行编辑：

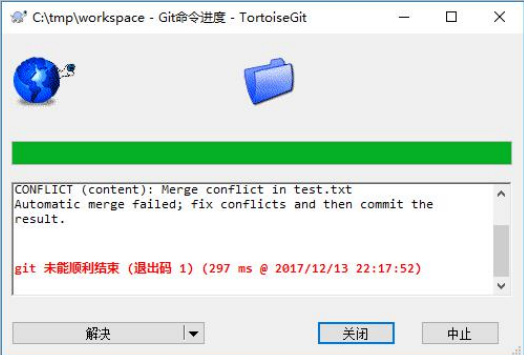


然后提交到版本库。

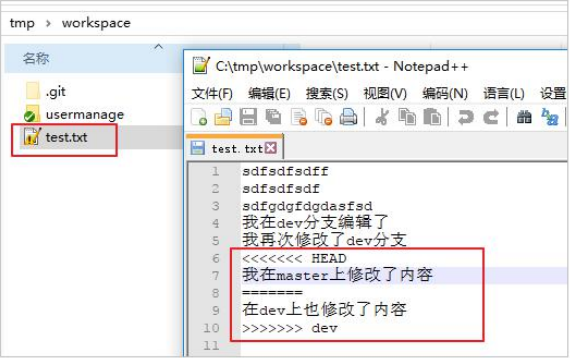
最后进行分支合并，例如将 dev 分支合并到 master 分支。需要先切换到 master 分支然后进行分支合并。



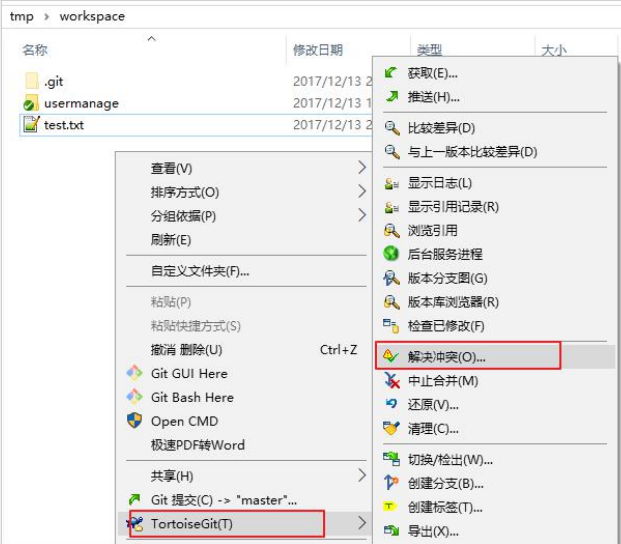
出现版本冲突。

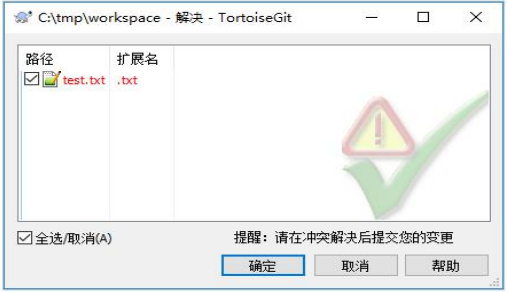


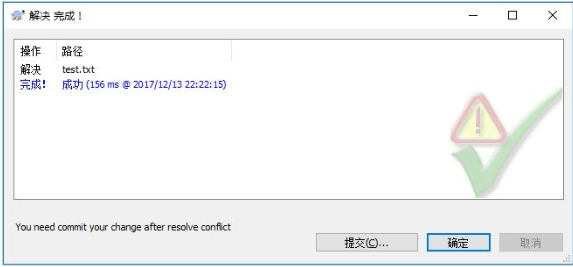
冲突需要手动解决，如下图解决完毕：



手动解决完冲突之后，在冲突文件上单机右键选择“解决冲突”菜单项：







把冲突解决完毕的文件提交到版本库就可以了。