# 版本控制

## 如何使用SVN版本控制

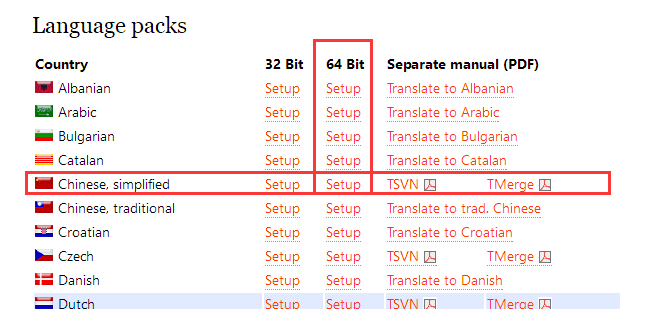
### 1.文件下载

tortoiseSVN1.9.6下载

<https://tortoisesvn.net/downloads.html>



tortoiseSVN1.9.6汉化包下载



Svn安装因权限问题导致的2503,2502的问题，解决办法如下：

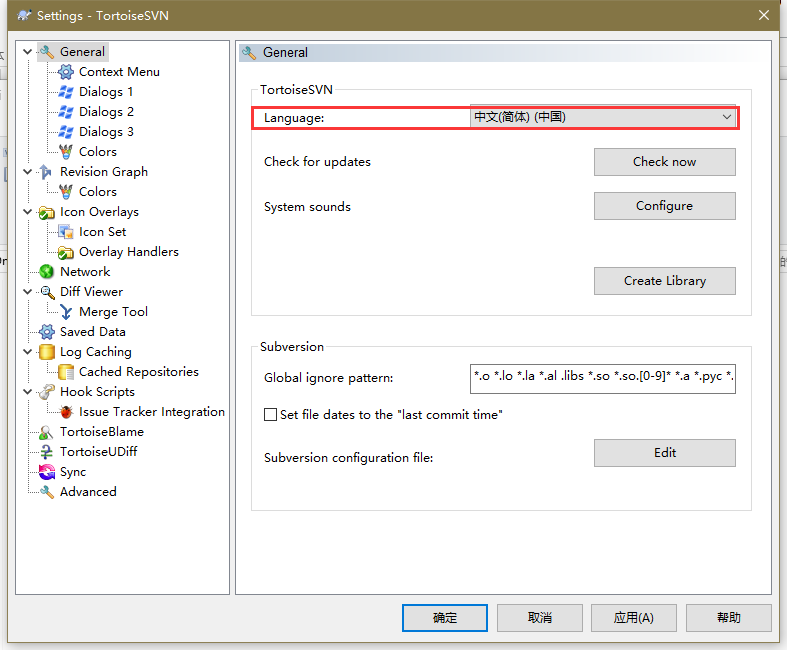
使用管理员命令打开cmd

**msiexec /package 你将要安装的程序的完整路径及程序名**

**svn先安装软件再安装汉化包**

### 2.语言设置

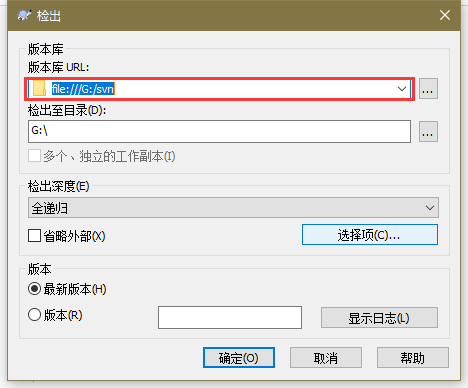
Svn安装成功之后，需要注销或重启电脑，鼠标右击tortoiseSVN选择Settings选项



语言现在中文，然后点击应用。

### 3.连接凭证

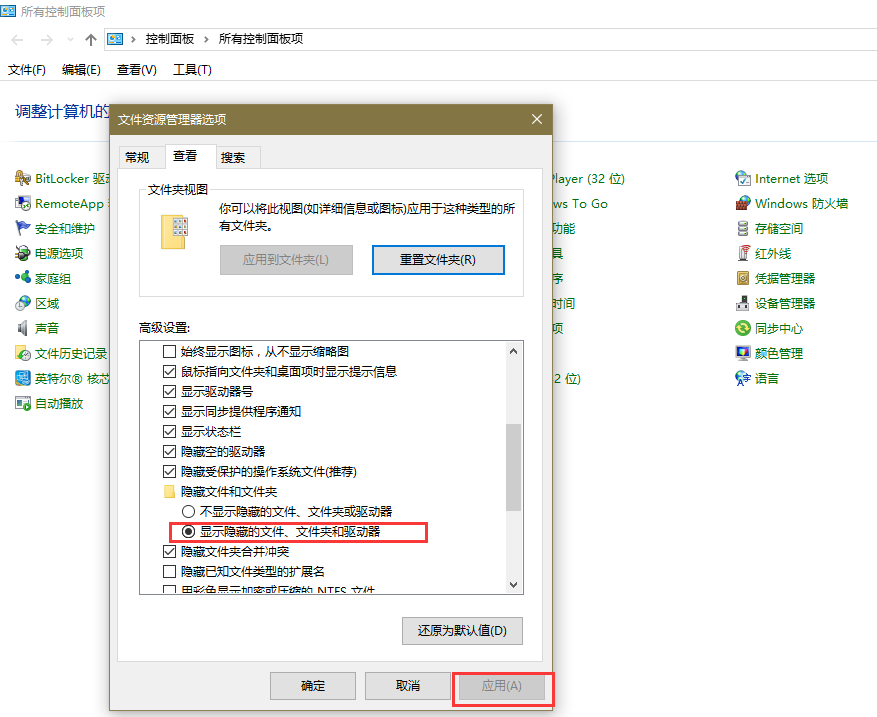
右击选择SVN检出，输入远端URL版本地址。以下本地默认



检出至目录是检出后保存的路径。点击确定后不一会就会弹出用户名和密码的对话框。

输入用户名和密码，并且记住即可。

第一次检出后，在控制面板中打开文件资源管理器，



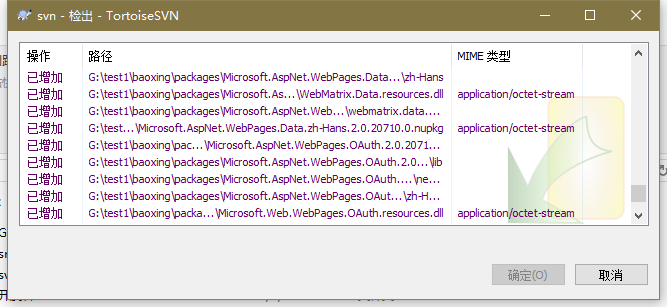
能看到更新后的svn隐藏文件夹。记录了当前更新的svn版本的信息，它位于检出的目录下



删除此文件夹后将会删除本地更新版本记录。

### 4.更新和检出版本

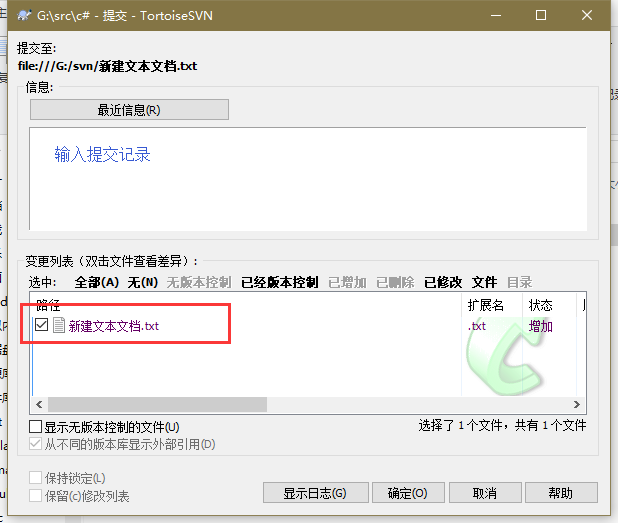
右击更新



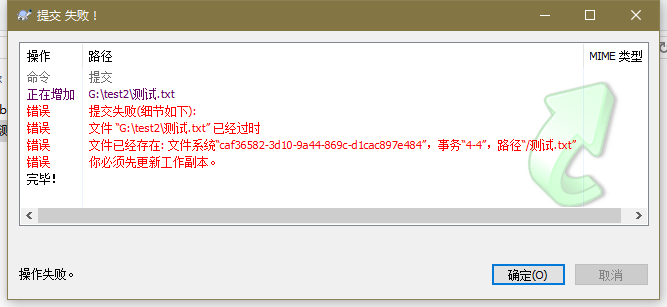
检出版本，选择右击检出。

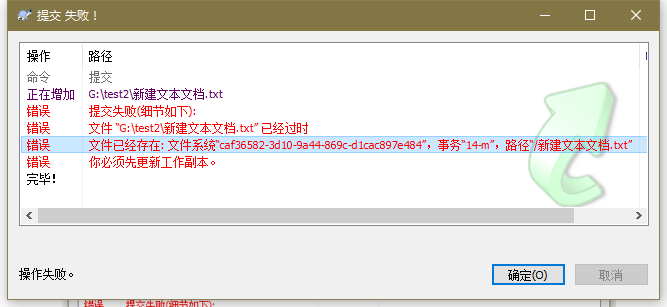
### 5.提交版本

输入提交记录，选择提交文件进行提交，也可以选择最近信息，添加提交记录



如果提交失败，如下图





则说明本地文件已经不是最新的，需要**先更新之后再进行提交**。

**所以提交代码的要求是，必须先更新后提交，避免冲突。**

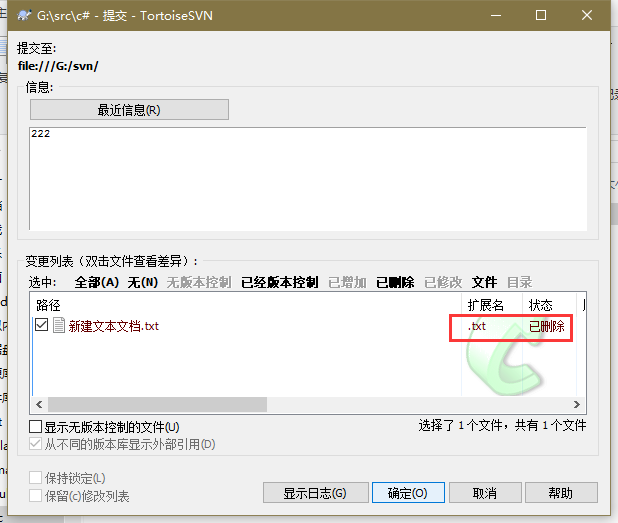
对于新建的文件一定要右击**添加**，之后才能进行提交

### 6.撤销版本

选择**项目或者文件**，右击**查看日志**，选择对应的**版本号**，右击。**复原此版本做出的修改,**

然后**重新提交，即可撤回版本。**

如果版本是新建的请选择上一级目录做复原操作



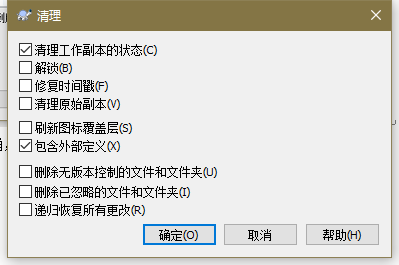
### 7.显示日志

可以输入作者/版本号/备注信息进行模糊搜索，可以查看历史版本的备注。

需要查看完整的历史版本，请点击下方的全部显示选项。



如果起始和结束日期不正确，应该是版本有问题。需要清理，选择项目右击清理

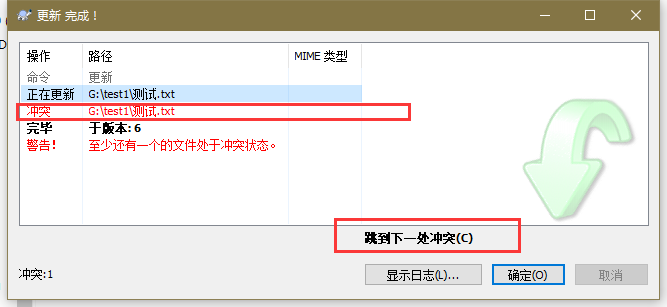


勾选清理的选项。比如修复时间戳。

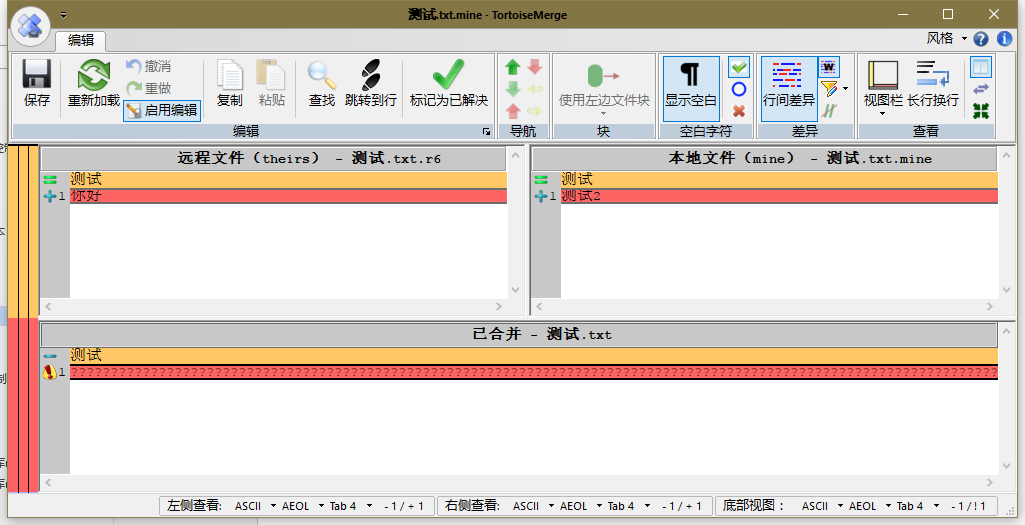
### 8.解决冲突

#### 文件冲突

更新文件时出现以下问题则出现了冲突。点击下一出冲突，可以查到列表有冲突的位置。

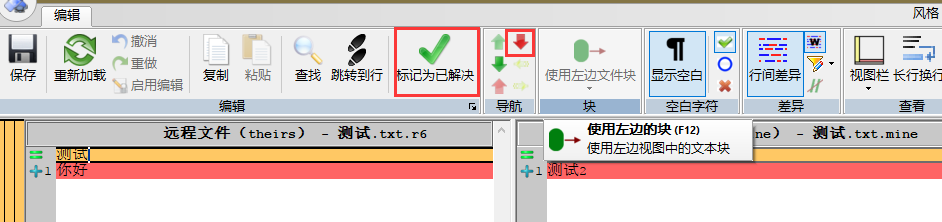


红色有冲突的部分，需要解决该冲突。选择文件右击，解决冲突。



选择冲突的代码，右击优先使用远程文件theirs或者优先使用本地文件mine，看实际需要。

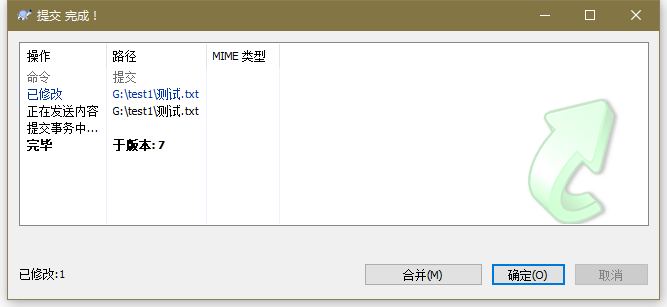
也可以在下面已合并的文件中直接手段修改。



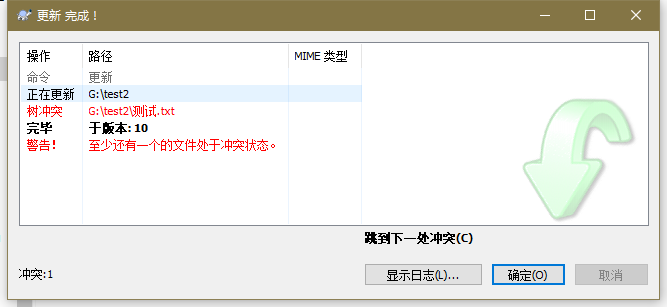
点击红色的下箭头可以找到该文件的下一个冲突。进行解决。

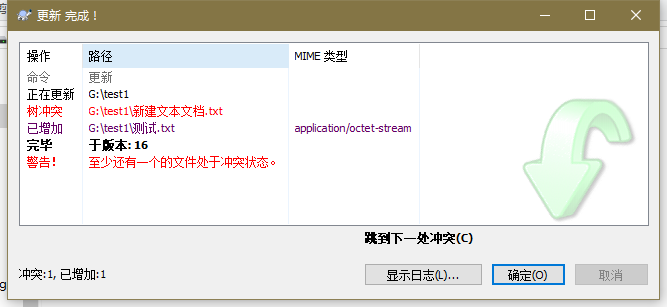
冲突全部解决之后，标记为已解决。

冲突解决完成之后再进行重新提交。



#### 目录冲突





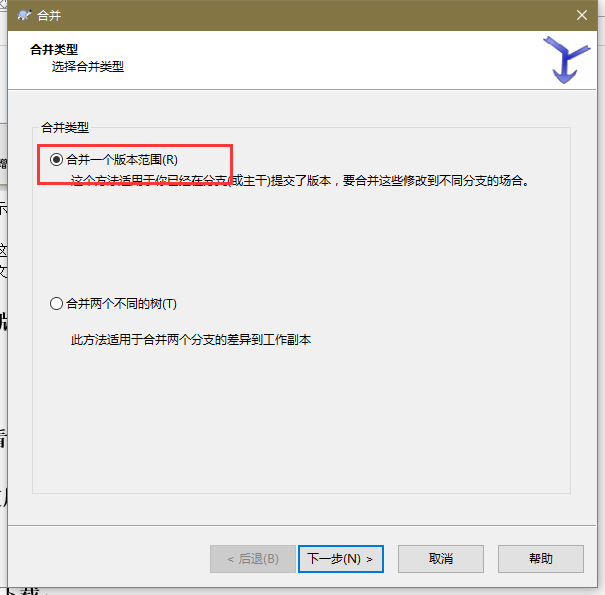
该冲突表示该文件已不存在G:\test1目录下了，需要先更新才能进行提交。

一般遇到这种情况都需要先找到删除该文件的记录，然后按照删除该文件的记录在本地项目中删除该文件然后进行提交。最主要还是先更新后提交。

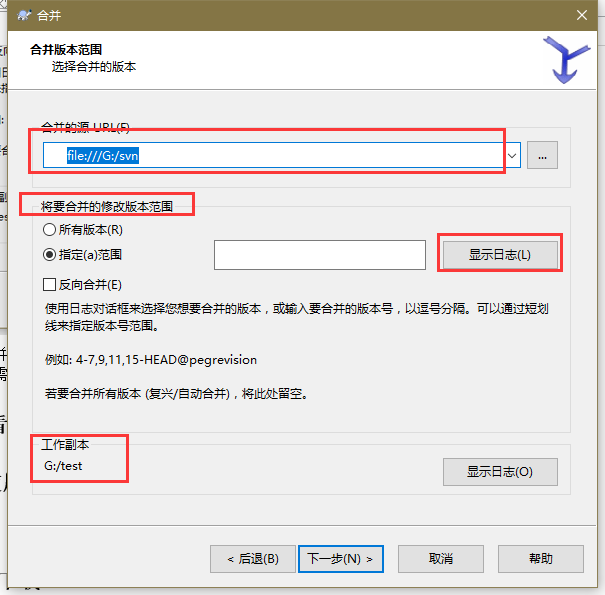
### 9.合并版本

选择项目，右击合并。

一般情况下将一个远端项目合并到另一个本地项目中，选择如下：

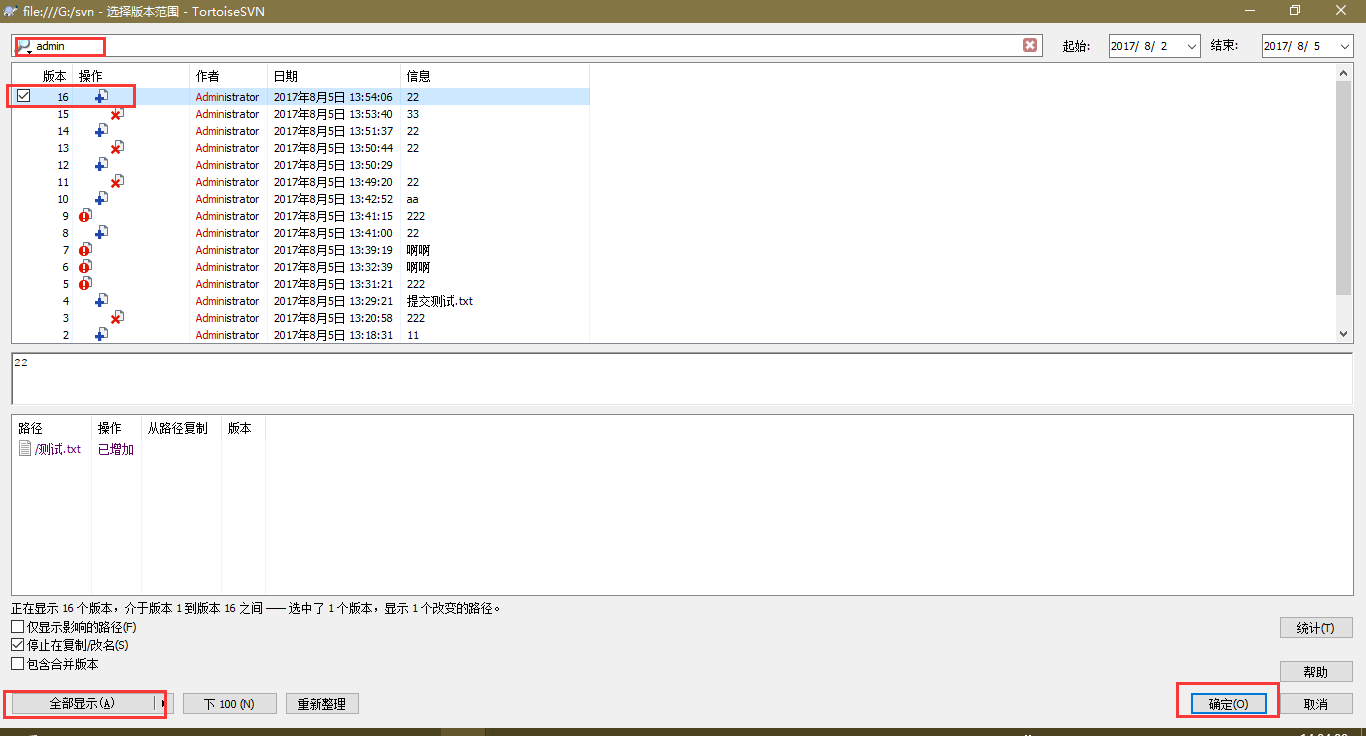


下一步



设置好合并的源:url,就是合并的源。本地就是工作副本。

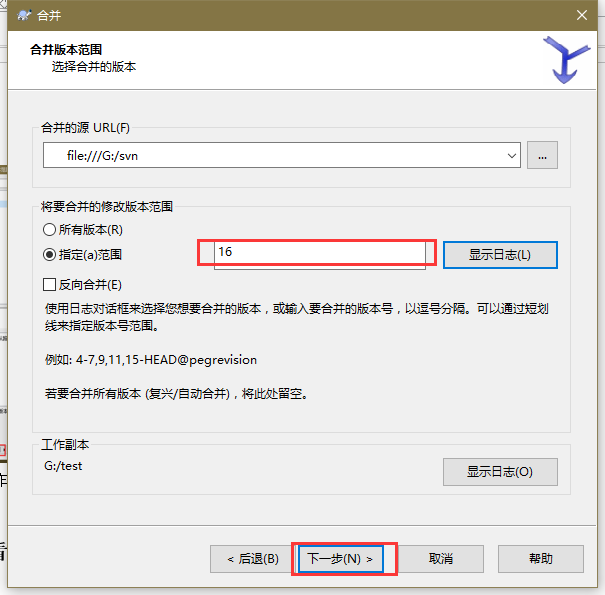
然后选择需要合并的版本范围。选择显示日志。

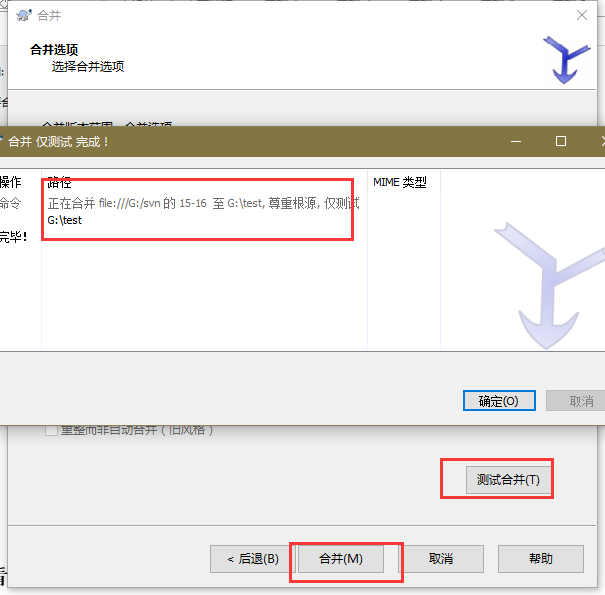


输入提交作者，选择需要合并的版本，如果需要选择历史的版本请选择全部显示。

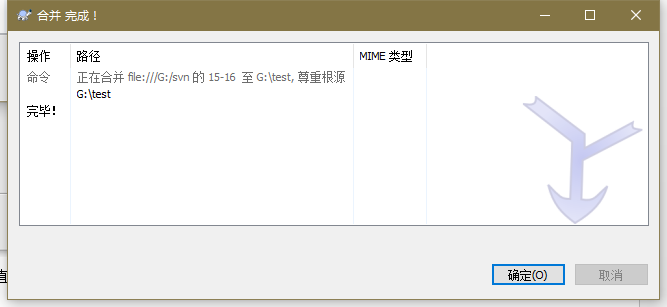
按照选择的版本要求：先合并版本小的再合并版本大的，就是自下往上合并。

选择好后，点击确定。选择下一步。





一般直接点击合并即可。



如果合并遇到文件冲突，则说明选择的合并版本不对。需要找到最先提交的版本才能提交后面的版本。

情况一：

比如说甲提交了”美好的生活.txt”文件,乙获取了甲的文件“美好的生活.txt”并进行了修改。然后提交。

然后合并时，乙较甲之前准备合并版本。乙合并时发现目录冲突。

然后乙找到甲，甲合并了自己的版本之后，乙再进行合并。合并成功。

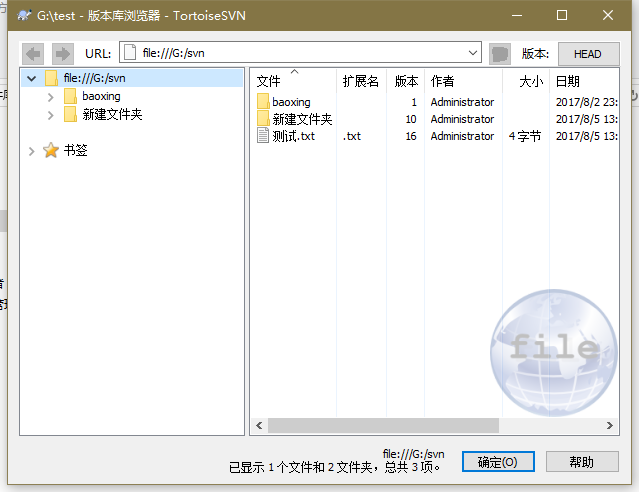
情况二：

甲提交了代码Calc方法。乙获取了甲的版本，调用了Calc方法。然后乙合并时没有冲突。

但是代码进行编译时，报错。提示Calc方法不存在。然后乙去找甲，甲不在。然后经过沟通，乙找到甲之前提交的版本好，先将甲的版本合并过去，再合并自己的版本。

### 10.查看远端版本

选择项目右击版本浏览器



## 如何使用Git版本控制

### 1.文件下载

1.下载git客户端:git-1.9.2-1401875809633.exe

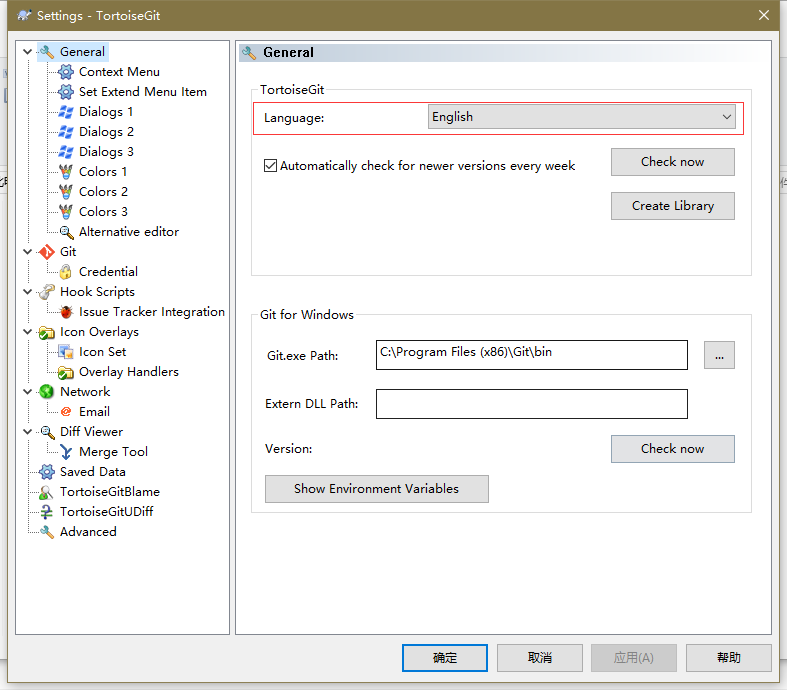
2.下载TortoiseGit-2.1.0.0-64bit.msi

3.下载TortoiseGit-LanguagePack-2.1.0.0-64bit-zh\_CN.msi汉化包

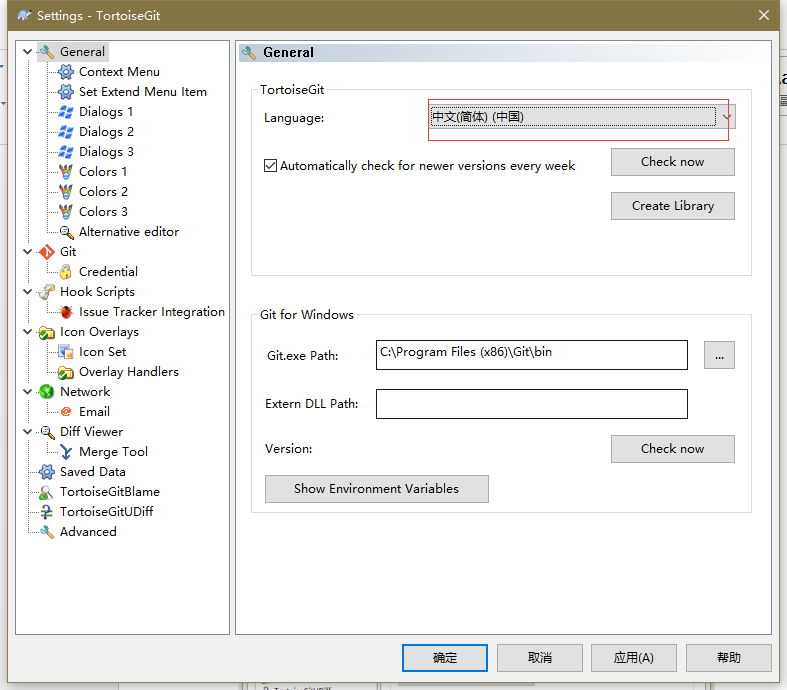
安装成功之后,注销电脑.

### 2.语言设置

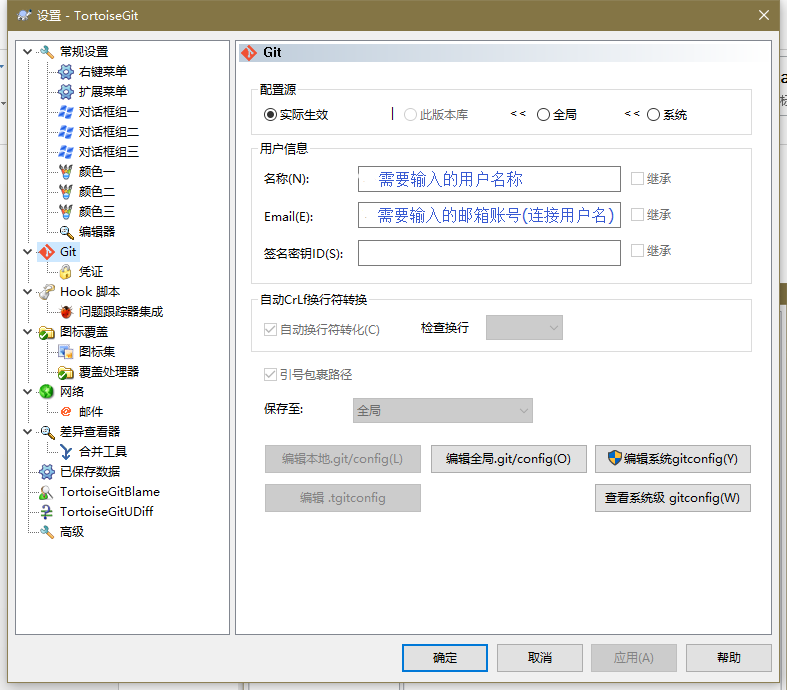
鼠标右击,显示快捷菜单，选择菜单项TortoiseGit的Settings,设置Language选项



选择中文后点击应用。



### 3.连接凭证设置



凭证助手选择**高级**

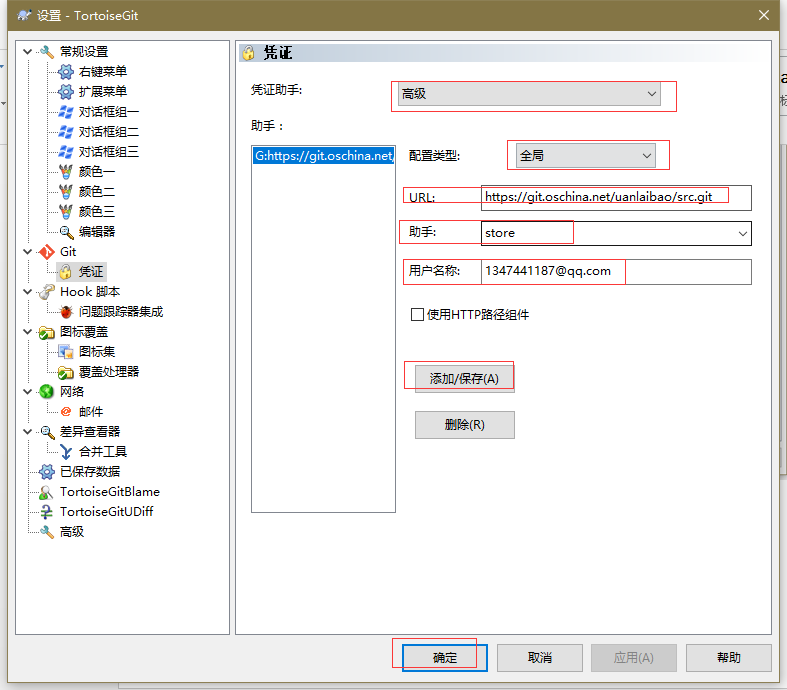
配置类型选择**全局**

URL**服务器的代码库路径**

助手选择**store**

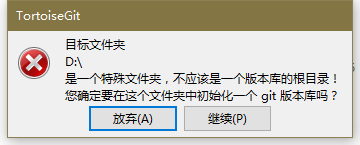
用户名称**登录的用户名**

输入完成后点击保存、确定



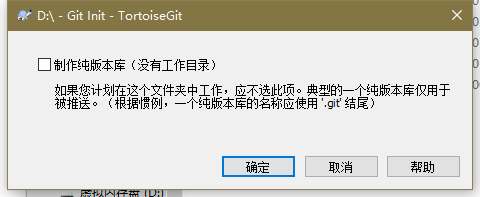
### 4.创建本地版本库master

鼠标右击,选择Git在这里创建版本库,



点击继续,

选择确定

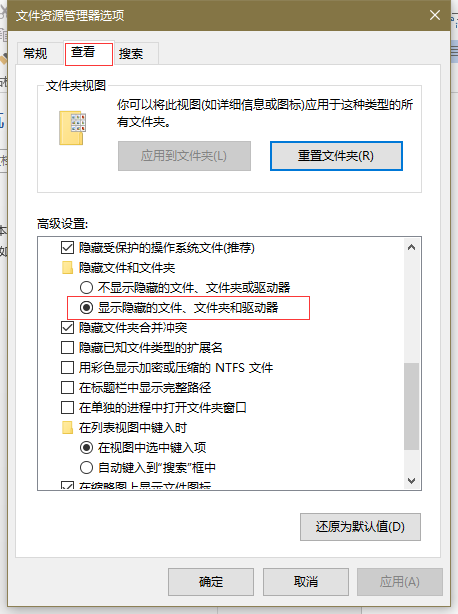


已经创建好了Git本地版本库了。

### 5.删除本地版本库master

打开控制面板

打开文件资源管理器，选择查看，显示隐藏的文件/文件夹/和驱动器



回到刚刚创建版本库的地方，显示了一个隐藏的文件夹名为.git



删除该文件夹，后设置文件资源管理器，不显示隐藏的文件/文件夹/和驱动器，

再次右击鼠标，发现可以重新创建版本库了

### 6.拉取版本

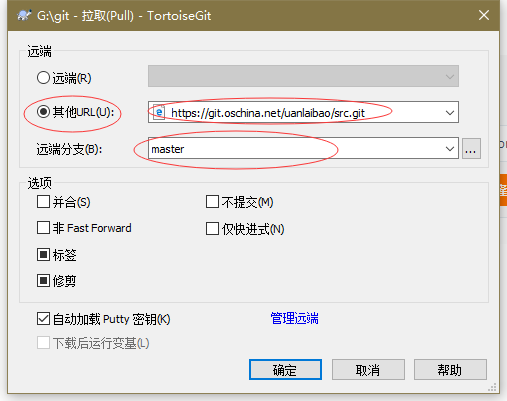
拉取版本是指第一次从服务器上获得文件。

获得服务器版本路径url,此处以码云为例，打开码云的项目，点击克隆或下载选项，获得git路径复制。



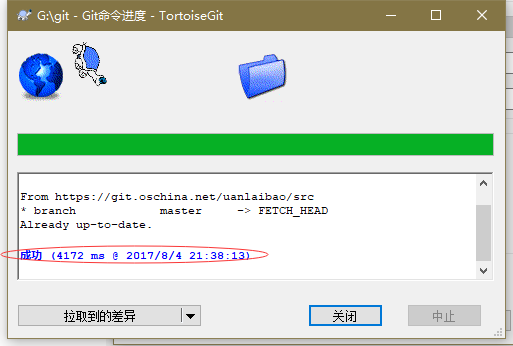
本地磁盘新建一个文件夹。

选择文件夹右击，选择**拉取**。



选择其他url,将**复制的url地址粘贴到此处**。远端分支默认是master.可修改。

完成之后点击**确定**。



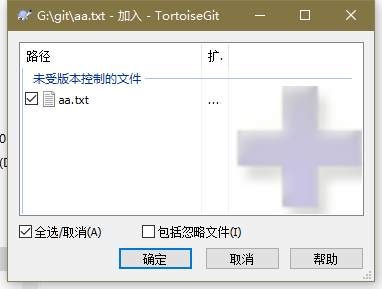
显示成功之后说明拉取版本成功。

### 7.提交版本

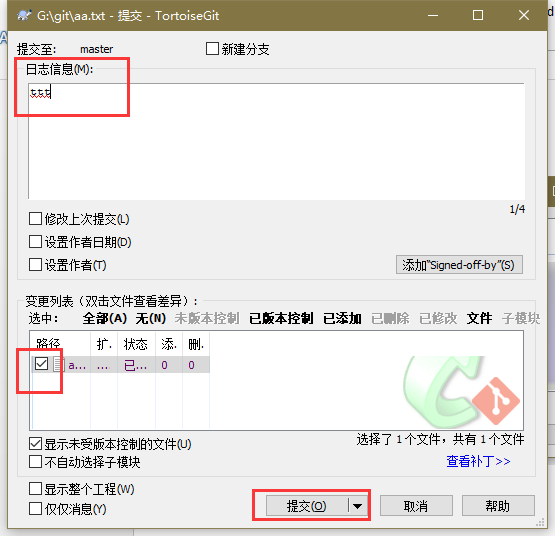
提交版本是指将当前修改的记录提交到本地版本库master

在版本库文件夹中**新建**一个记事本文件。

鼠标右击**添加**，如何点击**确定**

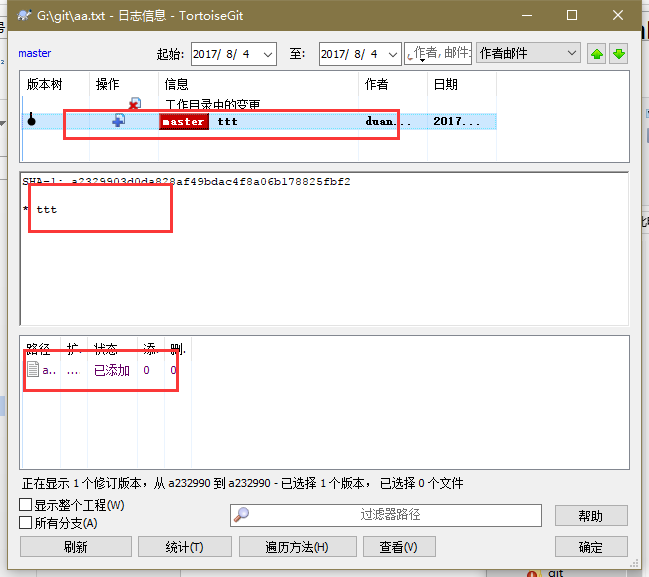


然后再次选择该文件，右击**提交**。



选择需要提交的文件，输入本次修改的日志信息，然后点击**提交**。

再次选择该文件，鼠标右击查看**日志**



可以看到提交人即作者，提交日期，备注，

回滚提交版本

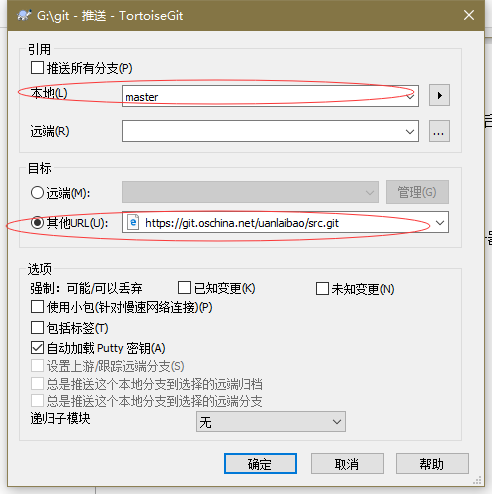
选择日志中的文件，右击可以看到“**还原到此版本选项**”。点击后可以还原。

然后将还原后的文件再次进行**提交**，实现了提交版本的回滚。

### 8.推送版本

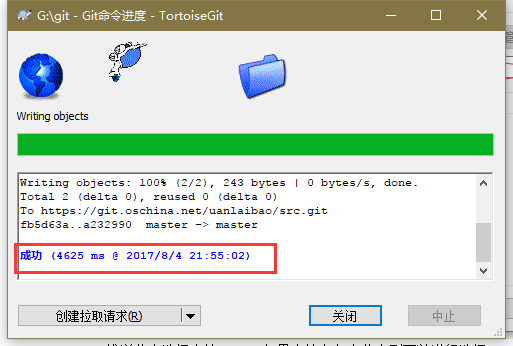
推送版本是指将本地版本库的修改记录推送到服务器。更新服务器上面的文件。

推送版本选择版本库根目录文件夹。鼠标右击**推送**



推送分支选择本地master,如果本地有多个分支则可以进行选择。

推送目标选择其他url输入服务器URL，点击确定

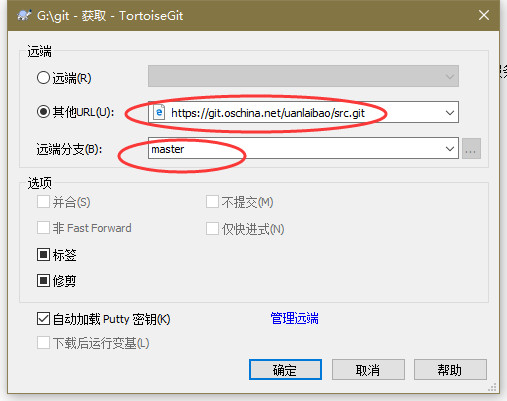


显示成功则推送成功，此时本地版本库提交的修改，将会更新在服务器上。

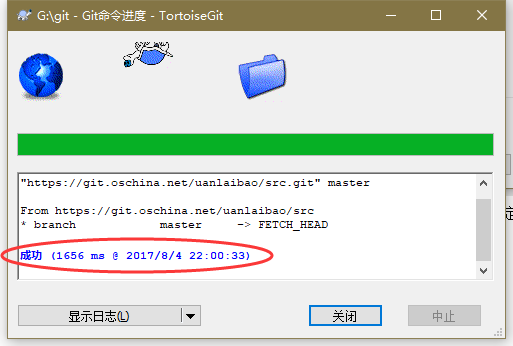
### 9.获取版本

获取版本是指本地版本库文件与服务器文件相比不是最新的，从服务器上下载最新文件到本地版本库。

右击选择**获取**。



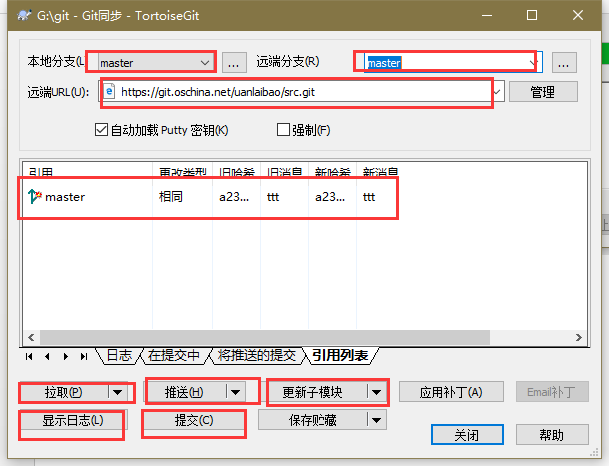
选择其他url,输入url地址。默认选择远端分支为master.选择后确定。



显示成功后，提示获取成功。

### 10.同步版本

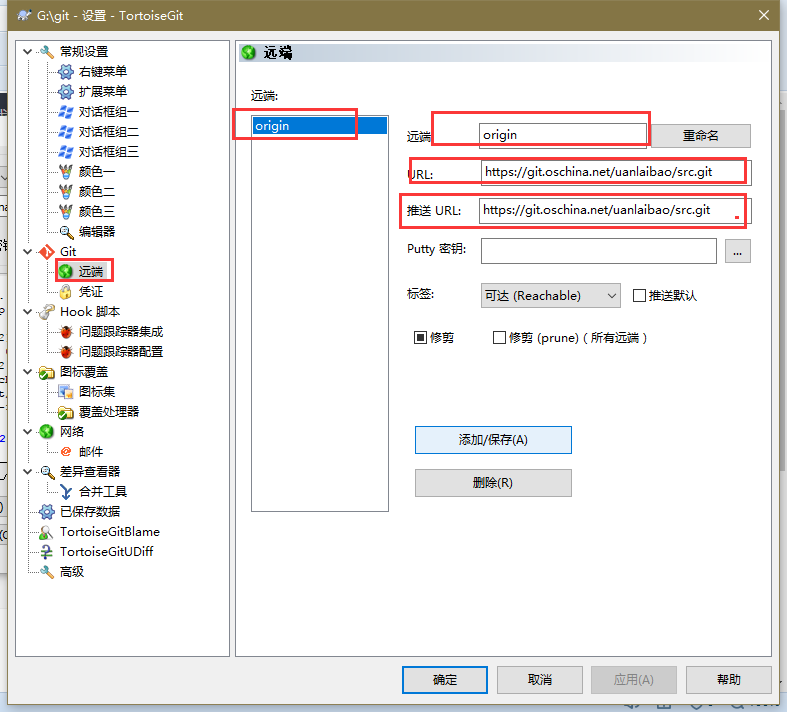
同步版本实现了本地分支与服务器分支的版本同步



选择本地分支，远端分支，输入远端URL.可以进行的操作有拉取、推送、提交

我们同步的流程是，先从服务器下载文件进行更新(拉取)，再提交文件到本地版本库分支(提交)，然后进行推送到服务器(推送)。

下面是远端URL配置：



### 11.创建分支

#### 创建分支

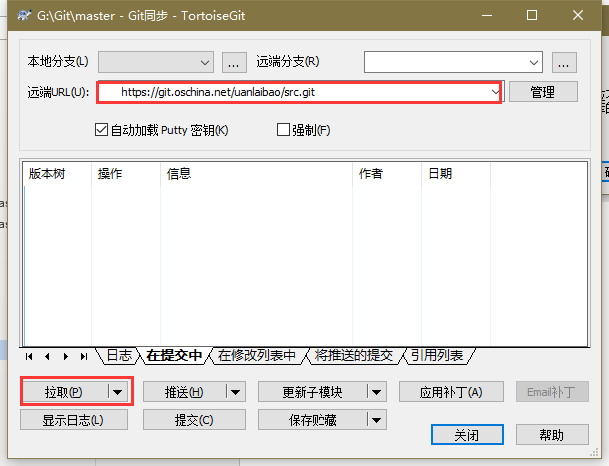
新建文件夹master



将master文件夹创建为本地版本库

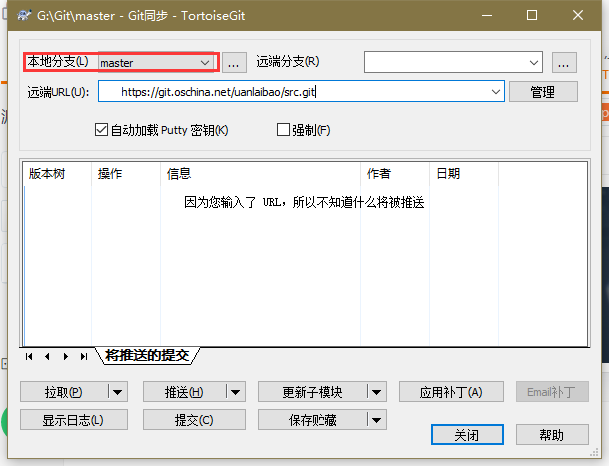


Master文件夹同步远端版本，输入**远端url**,然后进行**拉取**



拉取完成之后，该文件夹默认的分支就是master分支。选择master文件夹，右击选择同步，

此时本地分支就是master.

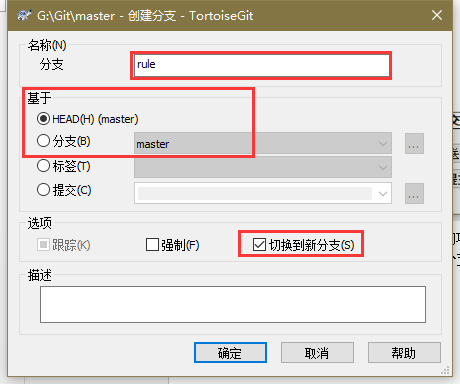


此时我们需要开发一个新的功能，但是不能直接提交到远端的master分支

#### 切换分支

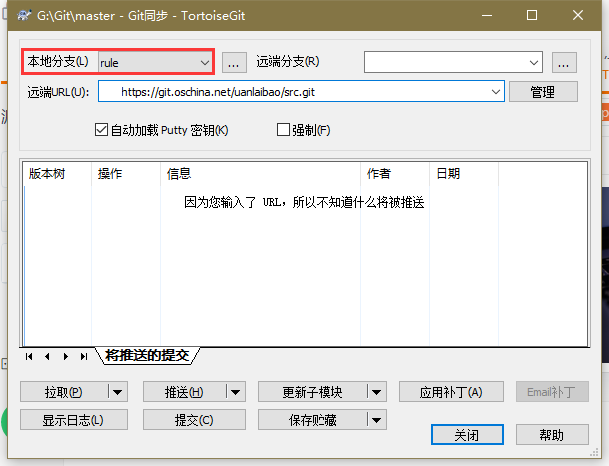
我们**选择项目右击，创建分支**。

输入分支的名称，选择**基于master**,勾选**切换到新分支，确定。**



此时我们再次选择项目右击，**同步**

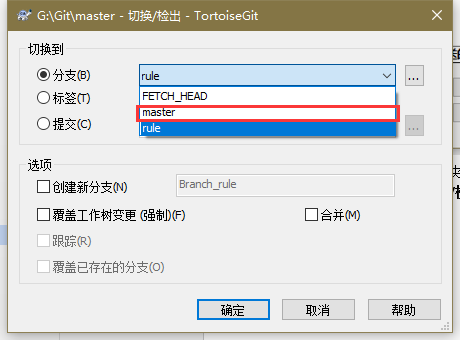
本地分支已经变成rule了



但是这个项目的文件夹是针对master的，我们需要将分支重新切回到master,

选择项目右击，**切换/检出**

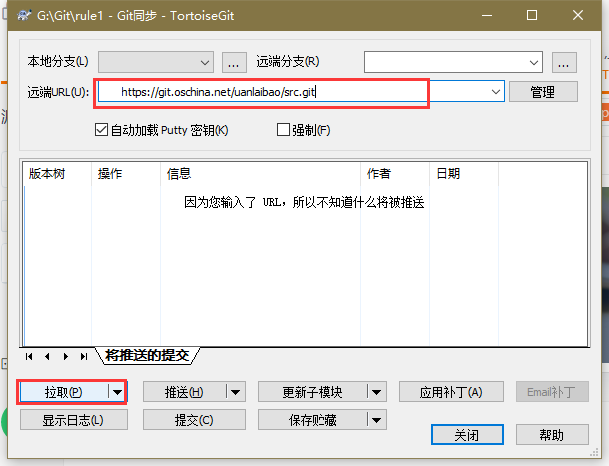
分支选择master，然后确定，就切换回master分支了



现在，我们再新建一个文件夹，名为rule1

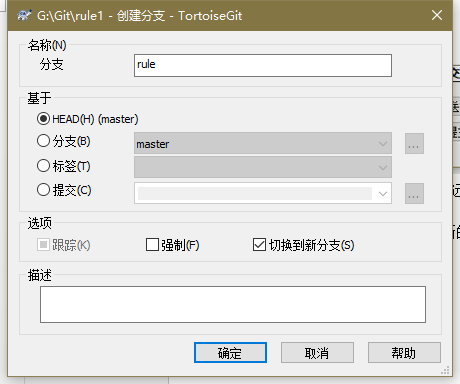


然后将rule1文件夹创建为版本库，选择同步，



更新rule1文件夹的版本与远端master版本为最新。此时rule1文件夹为master分支。

然后将该文件夹创建一个新的分支，分支名rule



此时rule1文件夹就属于rule分支了。

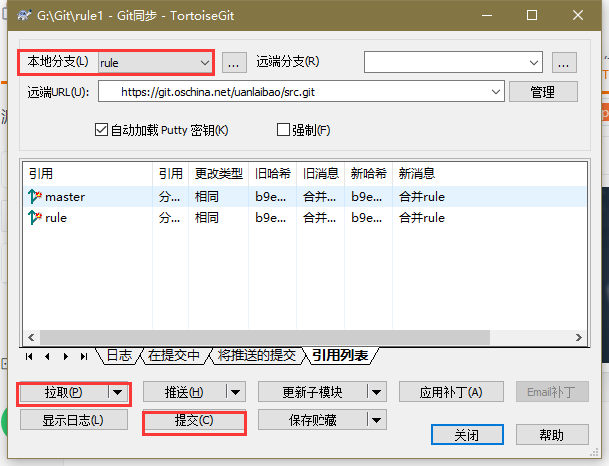
现在我们需要真正同步远端的rule分支，使其rule1分支真正属于rule的分支。

#### 同一分支不同位置提交

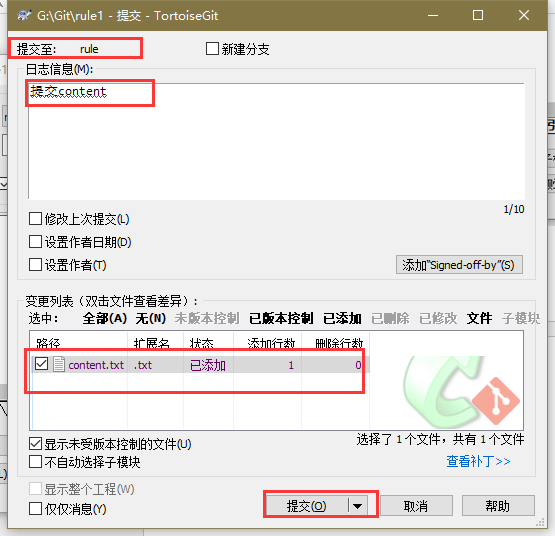
##### 先拉取>再提交>再推送

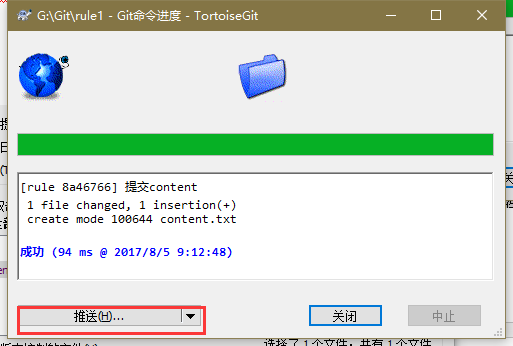
我们在rule1文件夹中新建一个文件，名为content.txt,右击添加。

再选择rule1文件夹(即rule本地分支根目录),右击选择同步

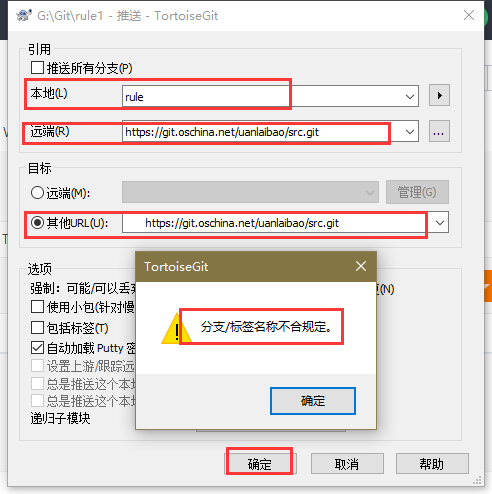


依此先**拉取**，后点击**提交,**输入日志和选择需要提交的文件，点击提交



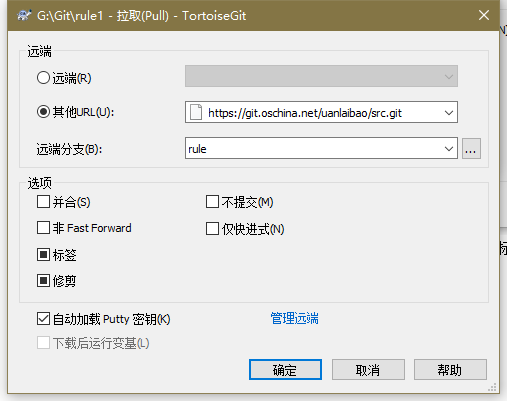


提交成功之后，点击推送，显示推送界面

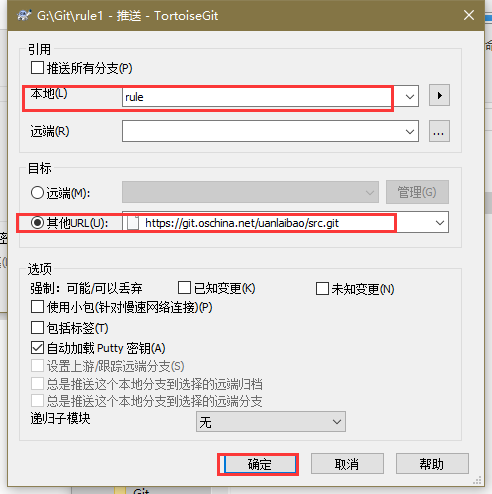


选择本地分支以及输入远端地址。然后点击确定。提示“分支/标签名称不合规定”,关闭提示信息和推送界面。

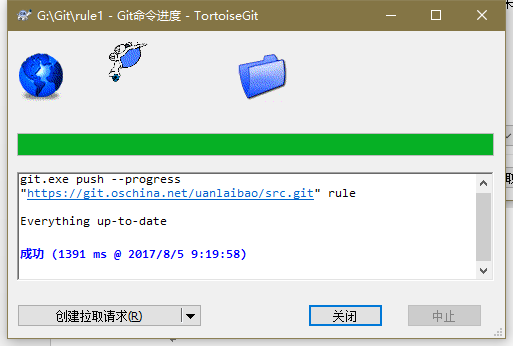
选择项目，右击**拉取**



拉取完成之后，再次选择项目，右击推送。



推送成功



##### 另一处分支的提交

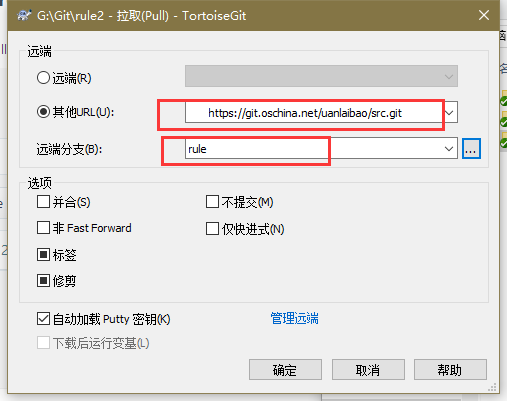
现在我们再新建一个文件夹，名为rule2,创建为版本库,然后拉取，默认是master版本库。

我们将rule2创建一个分支，名为rule.并且切换版本库为rule(切换/检出)

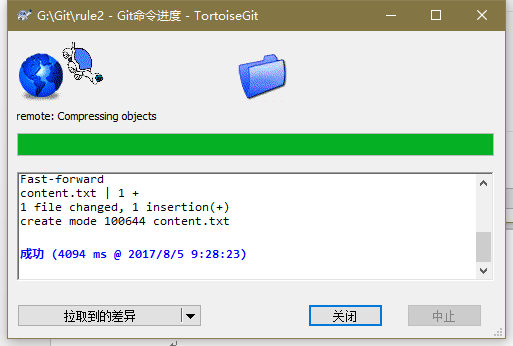
目前rule2里面只有master版本库的文件

现在我们要拉取rule远端版本库的文件content.txt,右击rule2文件夹，选择拉取

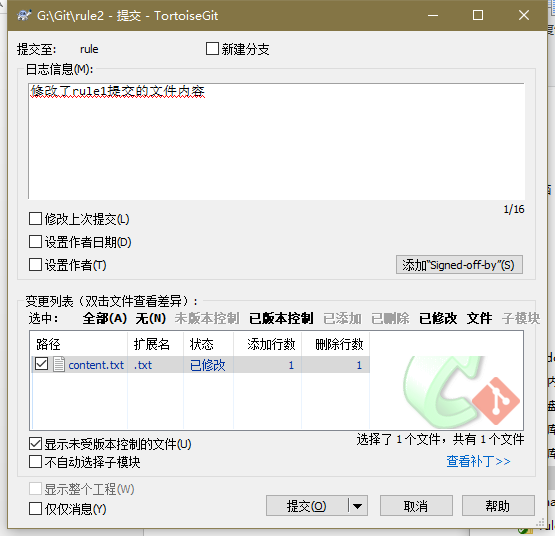
远端分支选择rule.



拉取成功



我们再将拉取到的新文件context.txt修改一下提交到rule远端

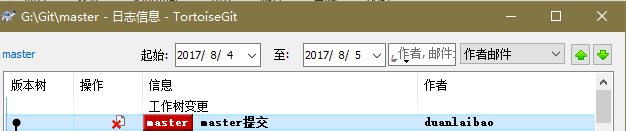


#### 分支日志

我们现在看看各分支的提交日志

首先我们将本地master文件夹拉取/本地rule1文件夹拉取/本地rule2文件夹拉取到最新代码。

Master版本库的显示日志只有master的提交记录



Rule版本库的显示日志有rule的2次提交记录



第1次提交了文件，第2此提交了内容。分别是不同地方的提交。

#### 同一分支的冲突

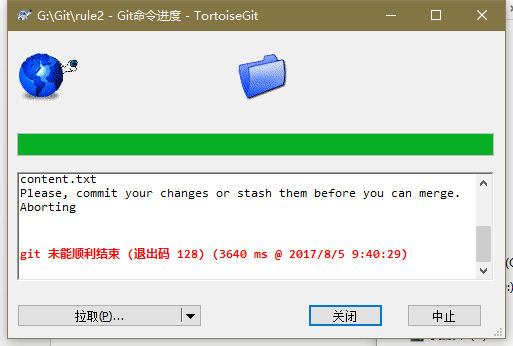
我们在rule1文件夹的content.txt文件里面写了1+1=2

我们在rule2文件夹的content.txt文件里面写了1+1=3

然后我们将rule1文件夹提交到rule版本库，然后进行推送.

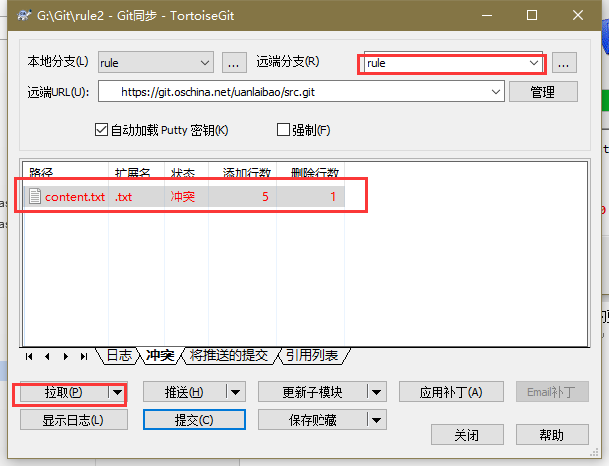
然后我们将rule2文件夹提交到rule版本库，

提交前，我们先拉取rule2为最新版本。



拉取失败。提示：Context.txt请提交您的更改或隐藏它们，然后才能合并。

我们选择项目，同步，再次进行拉取。

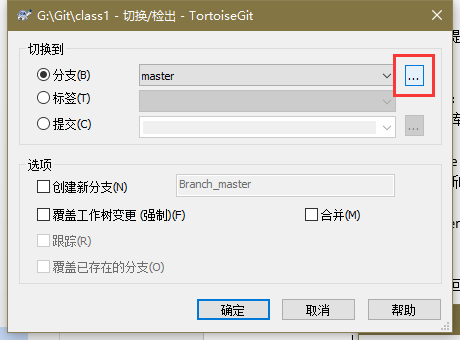


我们发现了该文件的冲突。右击**编辑冲突**。编辑冲突完成后**设置为已解决**。

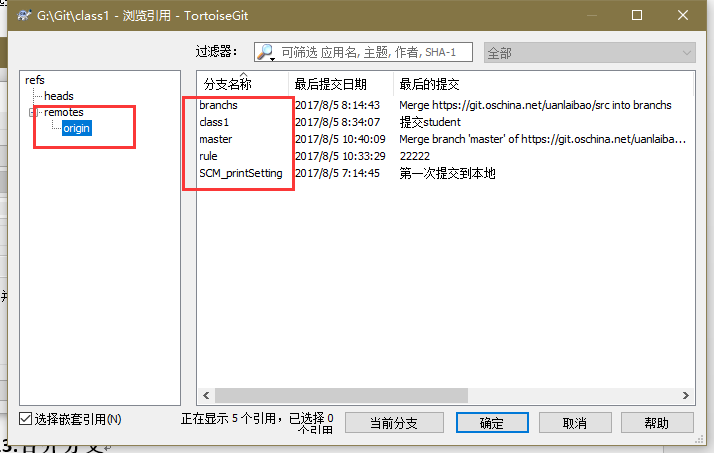
然后点击**提交，推送。**

### 12.删除分支

选择本地分支，切换到master主分支，右击切换/检出，然后选择更多



选择其中一个分支名称，右击删除



### 13.合并分支

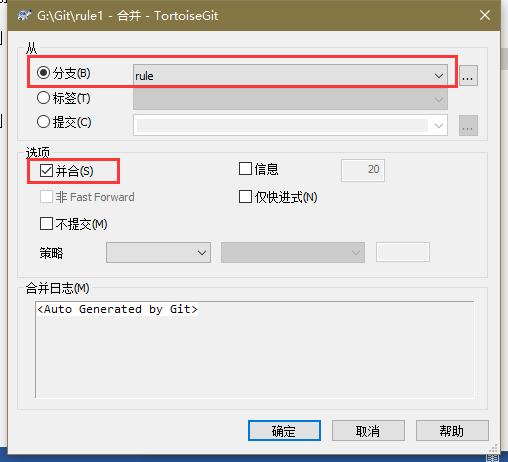
我们上面说了，如何提交代码并推送到云端。并且说了不同位置的分支版本提交。以及冲突解决。

下面我们需要讲的是：

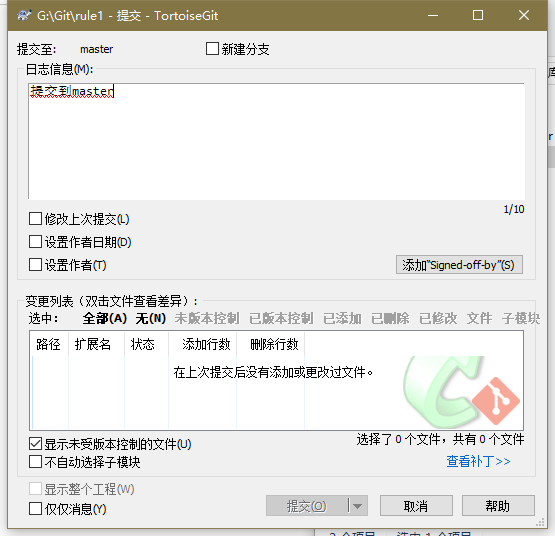
如何将rule分支版本库的文件合并到master分支版本库。

1. 我们选择一个rule分支(rule1文件夹)，先拉取,提交，推送，当rule分支与远端rule分支以及同步到最新时。我们**将rule分支**(rule1文件夹)**切换为master分支。**
2. 将已切换的master分支(rule1文件夹)，选择同步进行**拉取**到最新文件
3. 合并到master

当前文件夹已切回到master分支。然后合并时选择从**rule分支**进行**并合**操作



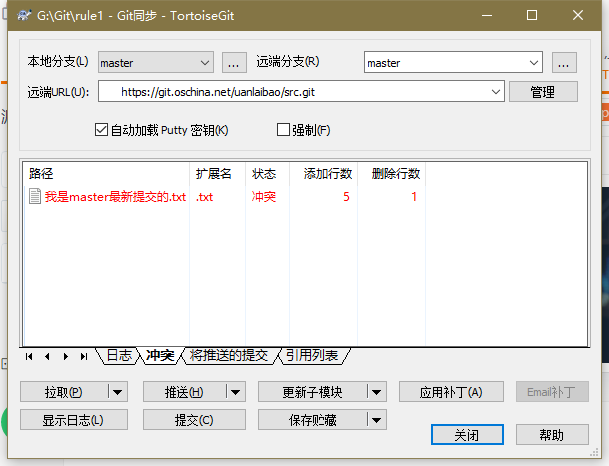
1. 提交到master



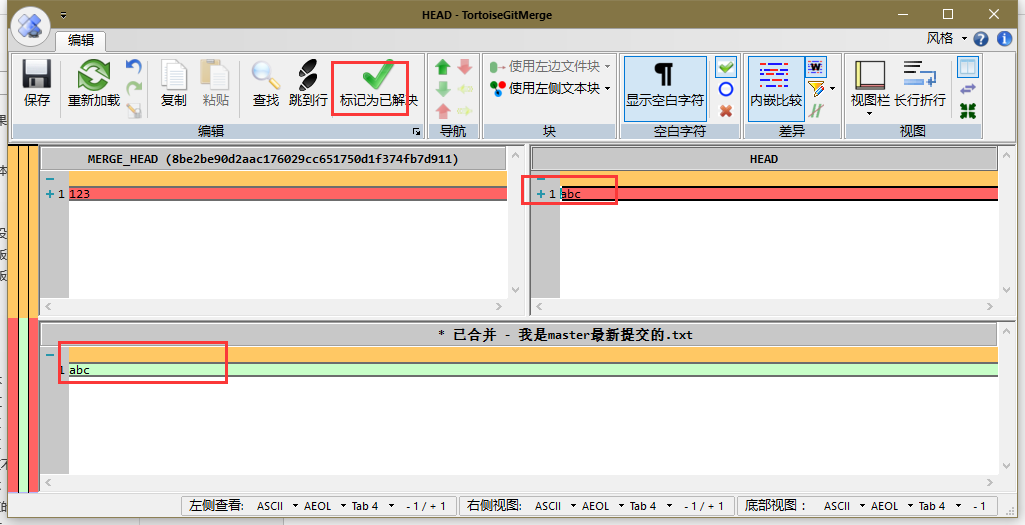
发现已无需再进行提交，然后进行推送。

#### 合并冲突

当rule1文件夹，属于rule分支时，我们切换分支到master,然后选择同步



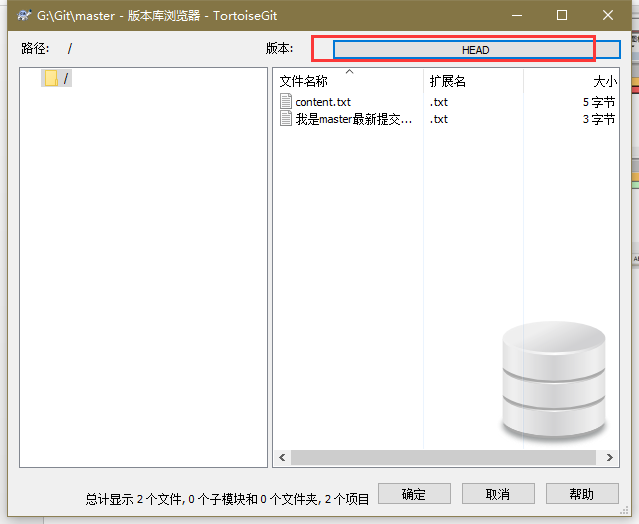
选择红色的文件，右击编辑冲突。然后标记为已解决。



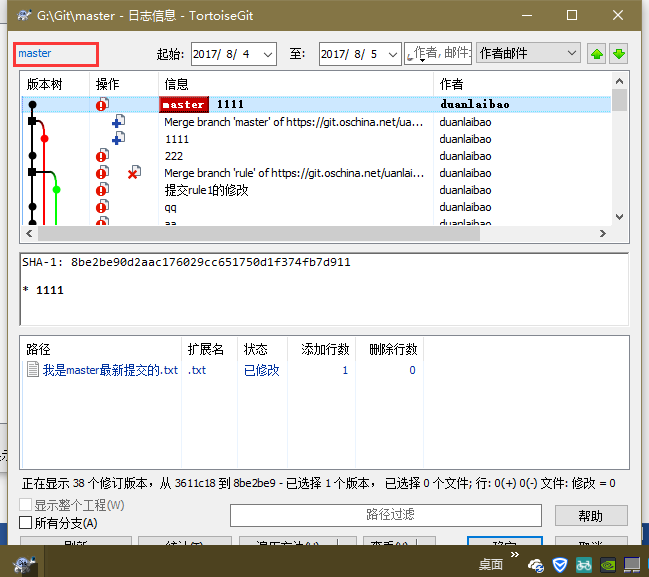
然后提交(如果是空则已空提交)，进行推送。

### 13.查看远端版本

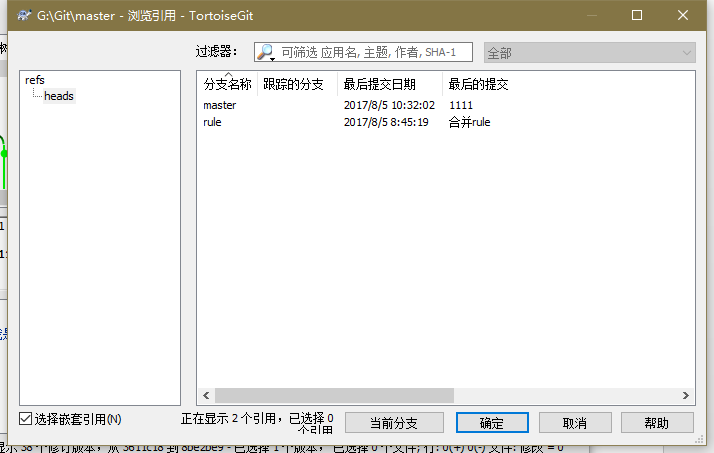
选择主分支master，右击版本浏览器，



选择版本HEAD,选择master分支



可以看到head里面有个分支分别是master和rule



### 14.查看本地版本

选择其中一个本地分支，右击选择版本浏览器,

