# TortoiseGit使用指南

目录

[TortoiseGit使用指南 1](#_Toc126143703)

[1 安装及配置 1](#_Toc126143704)

[1.1 下载安装 1](#_Toc126143705)

[1.2 初始化设置及Github SSH私钥配置 1](#_Toc126143706)

[2 存储库的建立和获取 8](#_Toc126143707)

[2.1 新建存储库 8](#_Toc126143708)

[2.2 获取Github远程存储库 10](#_Toc126143709)

[3 代码管理 13](#_Toc126143710)

[3.1 拉取代码 13](#_Toc126143711)

[3.2 提交代码 16](#_Toc126143712)

[3.3 推送代码 18](#_Toc126143713)

[4 分支管理 19](#_Toc126143714)

[4.1 创建分支 19](#_Toc126143715)

[4.2 合并分支 20](#_Toc126143716)

[4.3 删除分支 21](#_Toc126143717)

[4.4 其他 22](#_Toc126143718)

## 安装及配置

### 1.1 下载安装

首先确保Git已经安装，Git下载地址：https://git-scm.com/downloads/

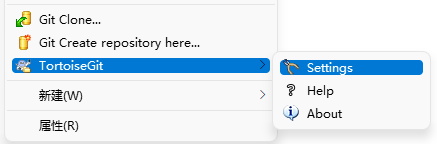
接下来安装TortoiseGit，它是Git的UI图形化插件，以图形界面取代命令行对Git进行操作。

下载地址：<https://tortoisegit.org/download/>

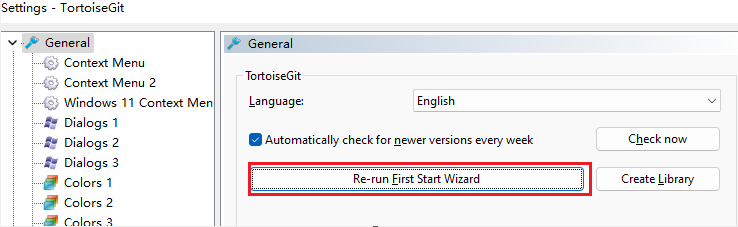
安装全程点击下一步。

### 1.2 初始化设置及Github SSH私钥配置

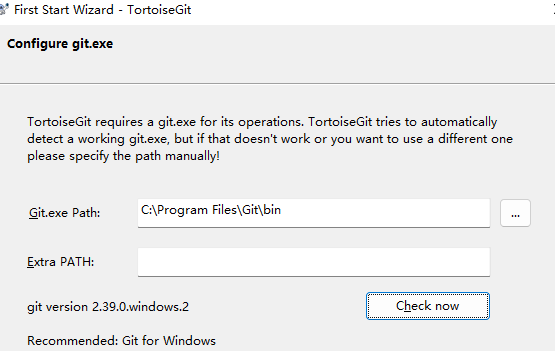
安装完成后在文件管理器右键打开设置界面。



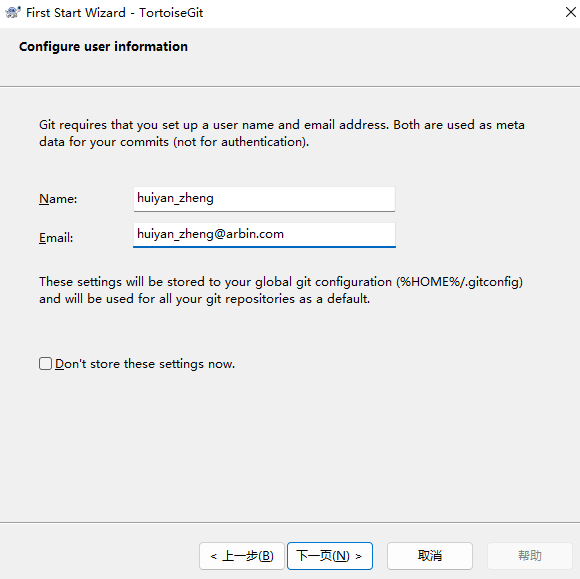
在General中点击初始化引导



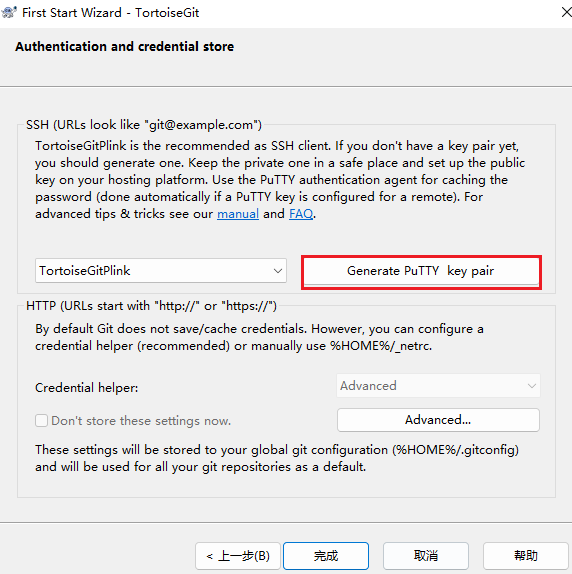
下一步到配置git.exe，点击Check now确认Git是否被识别



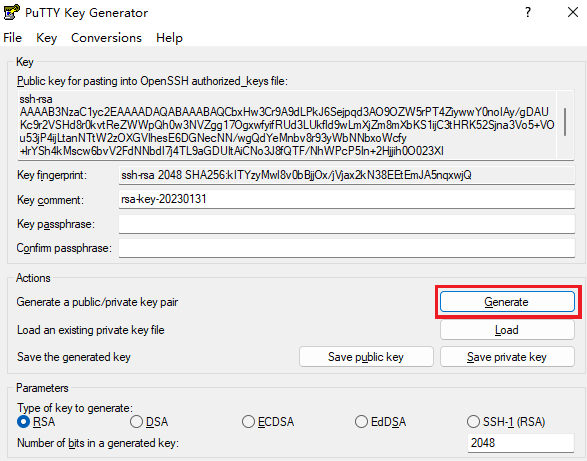
配置姓名和邮箱，此处配置影响Git的全局配置。



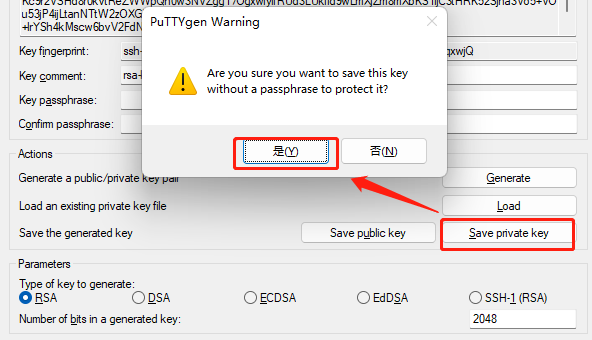
在下一页中点击生成PuTTY key，生成Github远端SSH凭证



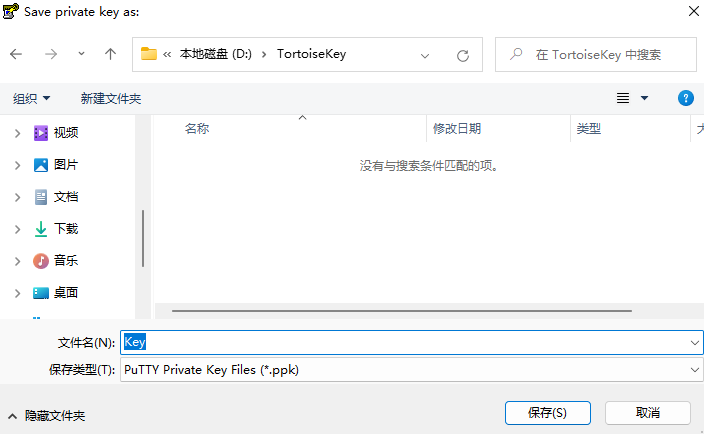
点击Generate，并在空白区域移动鼠标，会随机生成一段私钥



点击保存私钥，点击是。

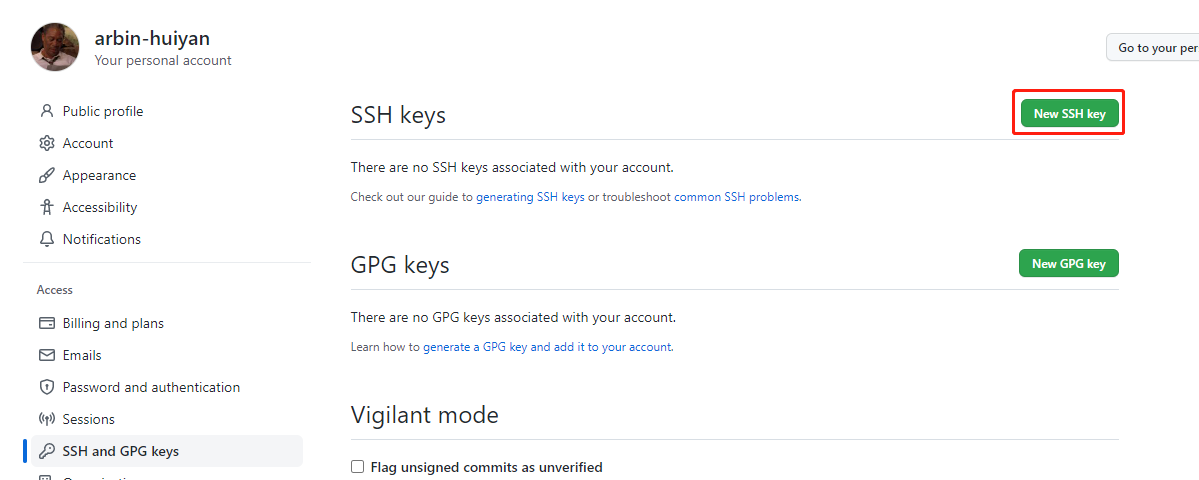


保存私钥到一个方便的位置。

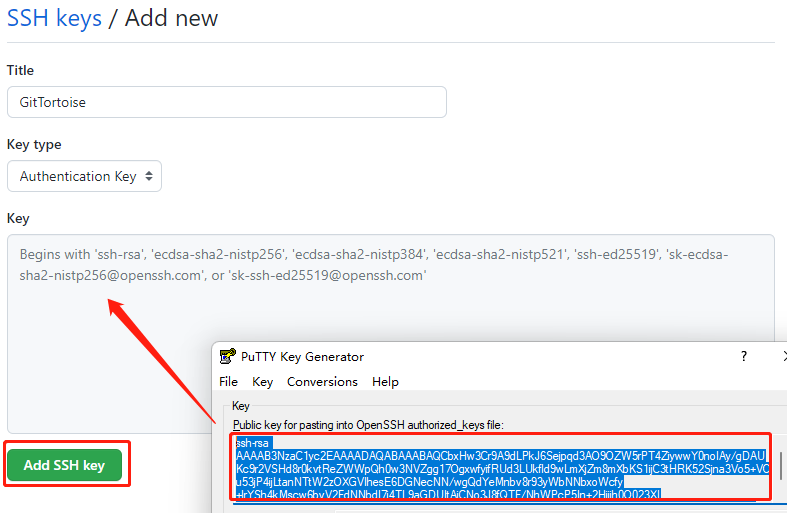


登录Github后点击右上角头像，进入Settings

在SSH and GPG keys栏目点击New SSH key

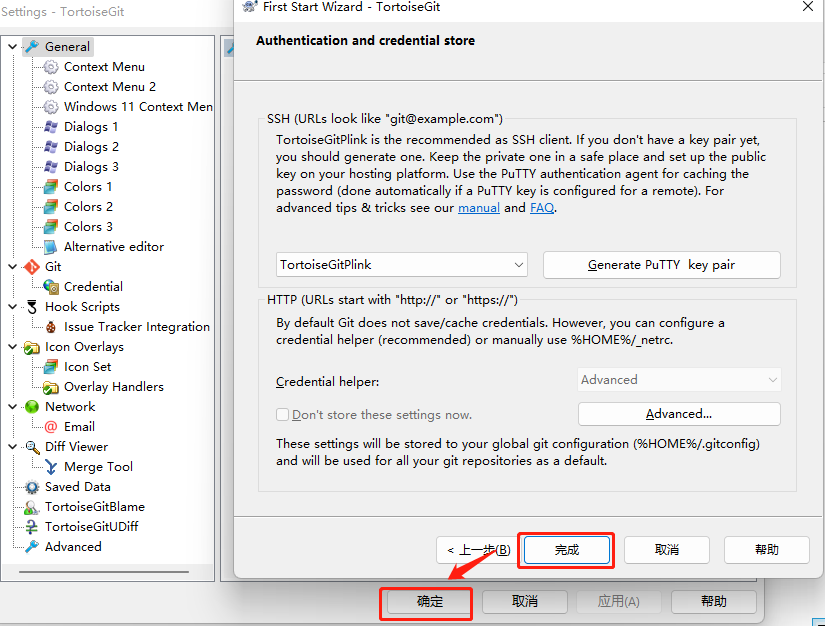


复制私钥到Key输入栏，点击Add SSH key



添加后，Tortoise可以通过该私钥访问GitHub远端存储库，而不需要输入账号密码。

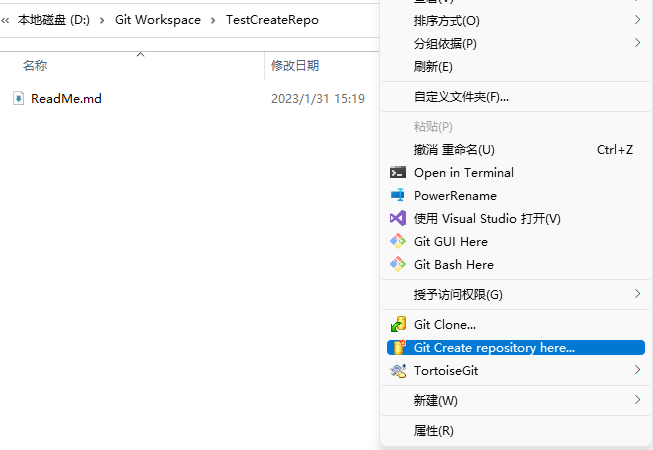
点击完成-确定关闭设置向导。



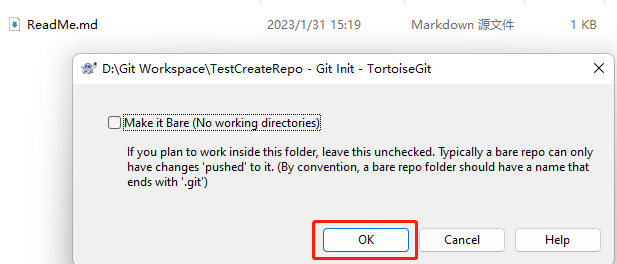
## 存储库的建立和获取

### 新建存储库

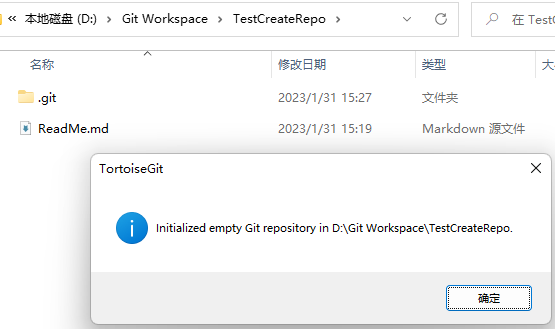
在需要新建存储库的目录下右键点击“Git Create repository here…”



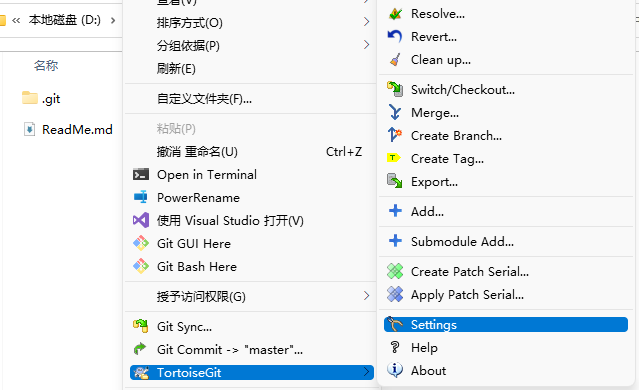
点击OK



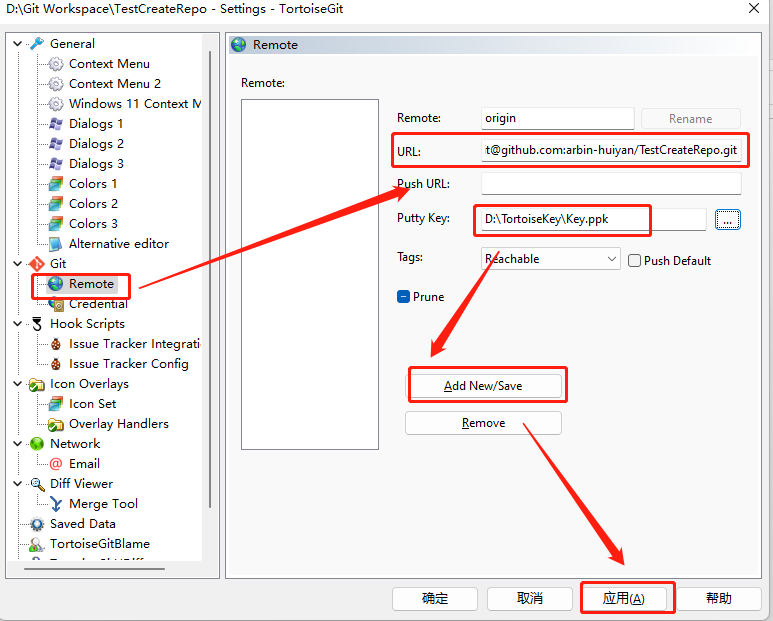
Git Init成功，生成.git隐藏目录



如果需要将该存储库与远程存储库进行匹配，方法是进入设置。



在Git的Remote中，添加远端地址。



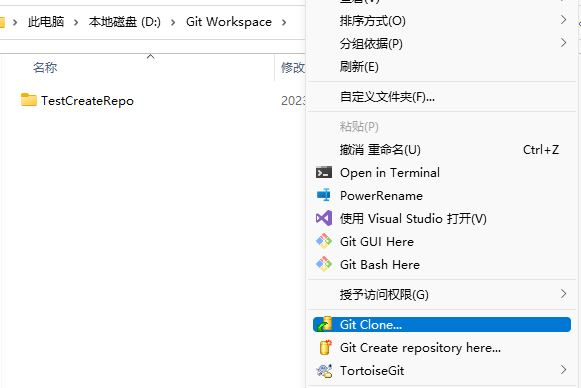
其中URL填写Github对应存储库的SSH地址

Putty Key选择之前在初始化设置中保存过的私钥，点击添加，再点击应用即可。

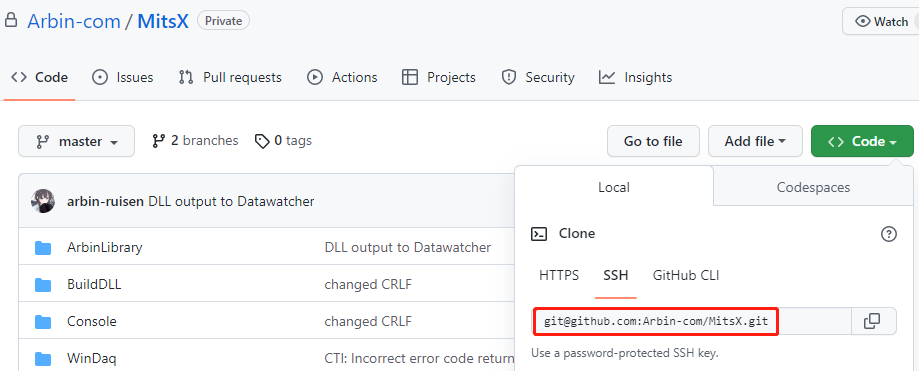
### 获取Github远程存储库

2.1中介绍了手动新建存储库并配置远程存储库的方法，也可以直接拉取远程存储库到本地——使用Git Clone

在存放的目录下右键点击Git Clone…

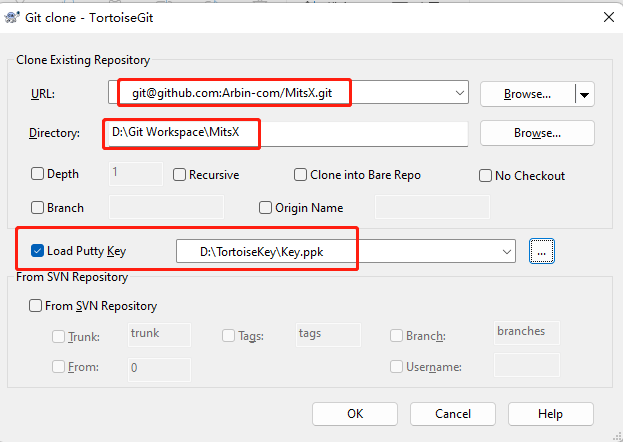


复制Github远程存储库的SSH地址

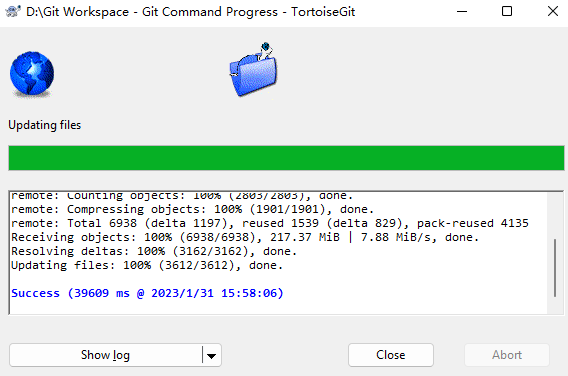


URL填写SSH地址，Directory为存放文件夹，勾选Load Putty Key，

选择在初始化设置中保存过的私钥，点击OK。

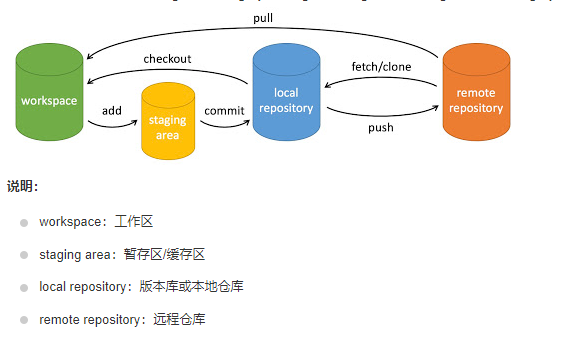


获取远程库成功。

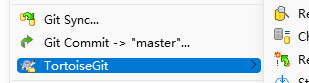


## 代码管理

Git 常用的是以下 6 个命令：git clone、git push、git add 、git commit、git checkout、git pull



通过文件资源管理器进入对应的项目目录，右键文件夹、空白区或者文件进入TortoiseGit子菜单来执行想要的Git命令，这就需要对Git命令的作用有一定的了解。



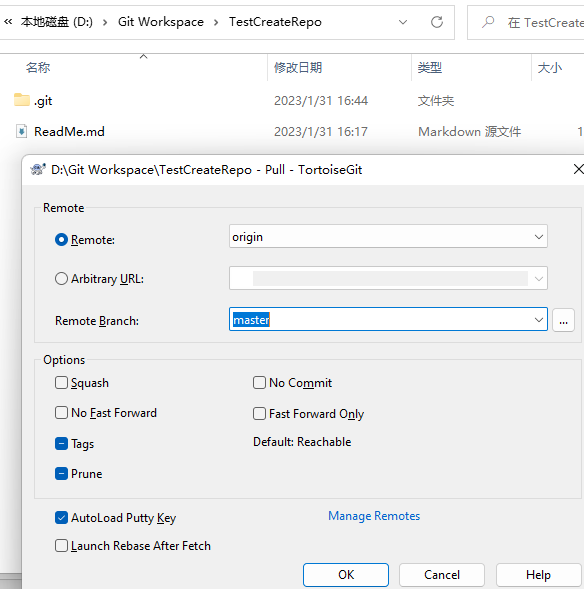
### 拉取代码

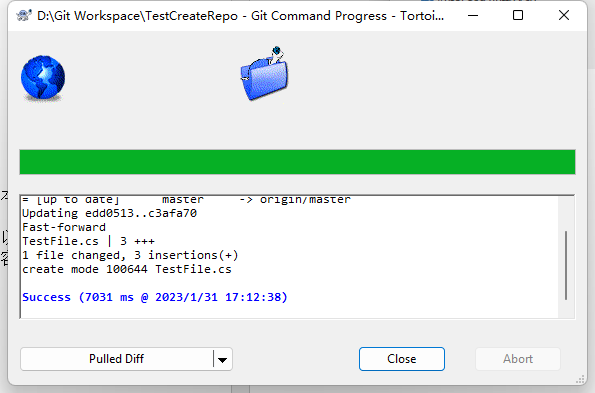
2.2中介绍了通过Git Clone来拷贝一个 Github远程存储库到本地。那么，拉取远程仓库最新代码可以通过Fetch/Pull实现。

git fetch 是将远程主机的最新内容拉到本地，用户在检查了以后决定是否合并到工作本机分支中。 而 git pull 则是将远程主机的最新内容拉下来后直接合并，即： git pull = git fetch + git merge。

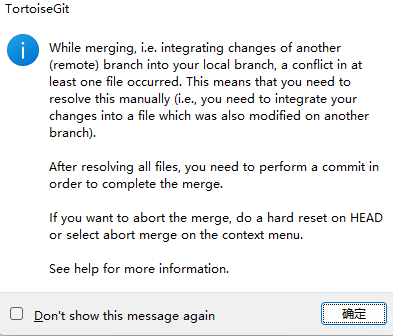
一般我们拉取代码直接使用git pull命令。

点击右键菜单-TortoiseGit-Pull按钮，选择Remote（选择设置好的远程库地址别名，一般为origin），和Remote Branch（远程分支），点击OK，即可拉取远程库最新内容。

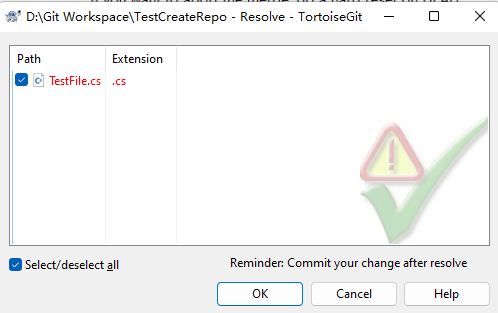




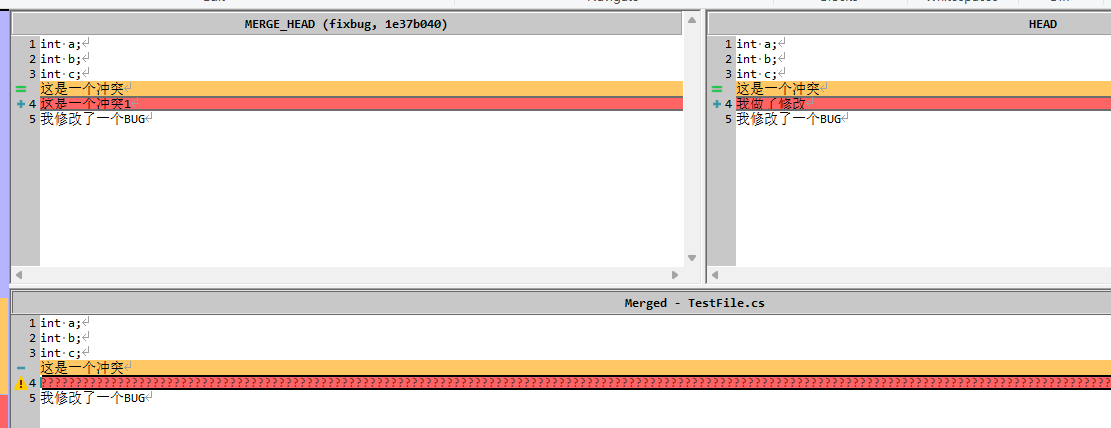
如果Git Pull后与本地库有冲突，会出现提示，需要手动处理冲突，并进行Commit操作解决冲突，如果没有处理冲突，Commit弹窗中也会显示冲突内容并进行警告。



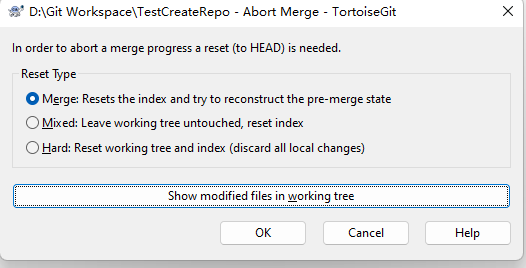
进入菜单项-Resolve…解决冲突



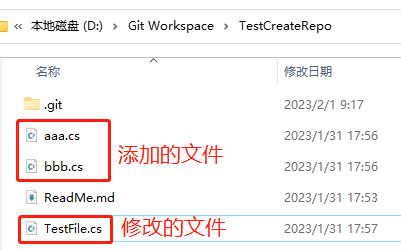
双击冲突的文件后，进入文本编辑器，可以手动处理冲突，处理完成后，就可以进行Commit和Push操作。



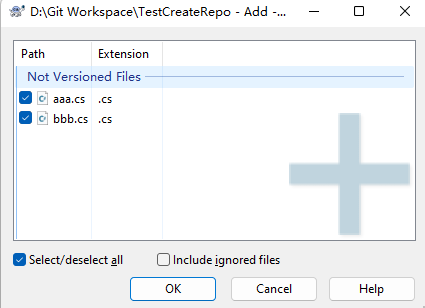
或者进入菜单项-Abort Merge用于回退版本



### 提交代码

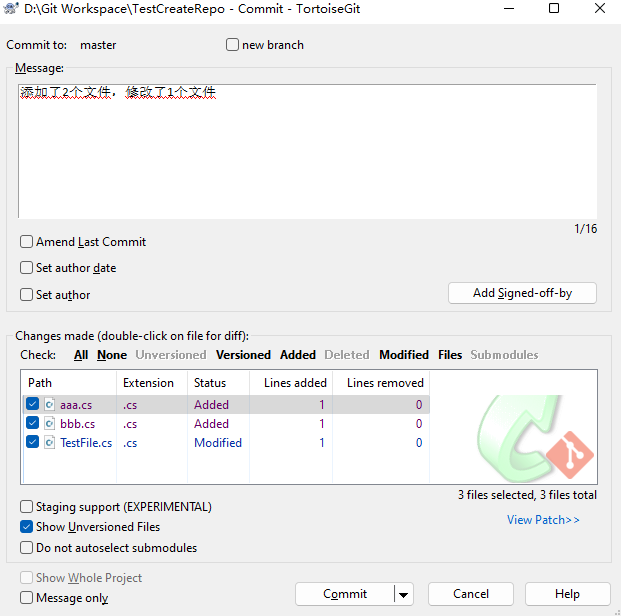


执行Git Add添加文件到暂存区。



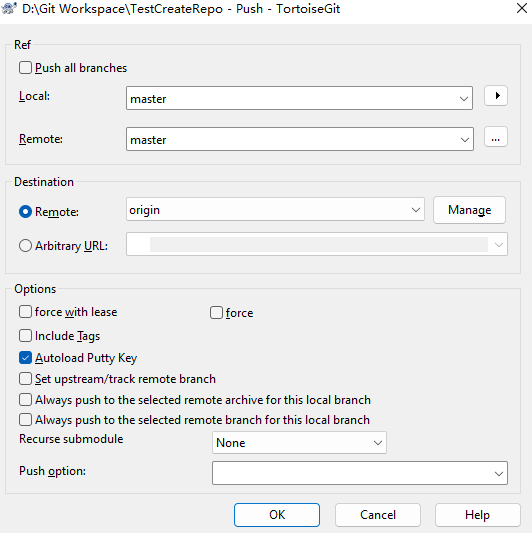
执行Diff可以查看与暂存区的区别

执行Git Commit



### 推送代码

执行Git Push



Local选择本地分支，Remote选择远程分支

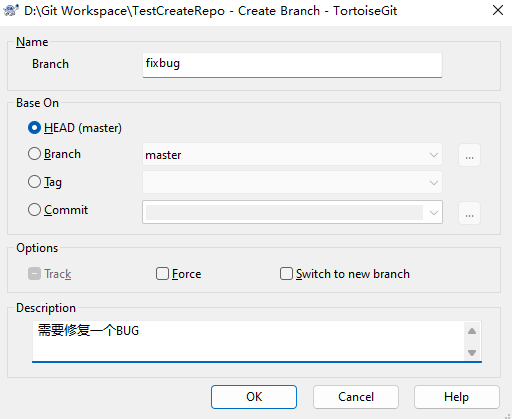
勾选force--会使用本地分支的提交覆盖远端推送分支的提交。

勾选了force-with-lease 之后，使用此参数推送，如果远端有其他人推送了新的提交，那么推送将被拒绝，这种拒绝和没有加 --force 参数时的拒绝是一样的。

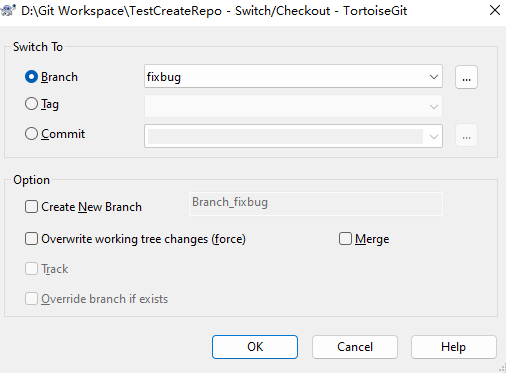
## 分支管理

### 创建分支

执行Create Branch



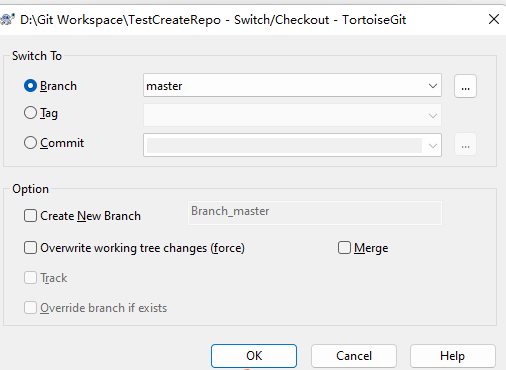
Switch/Checkout到新建的分支，也可以在这里直接勾选Create New Branch新建并切换一步完成。



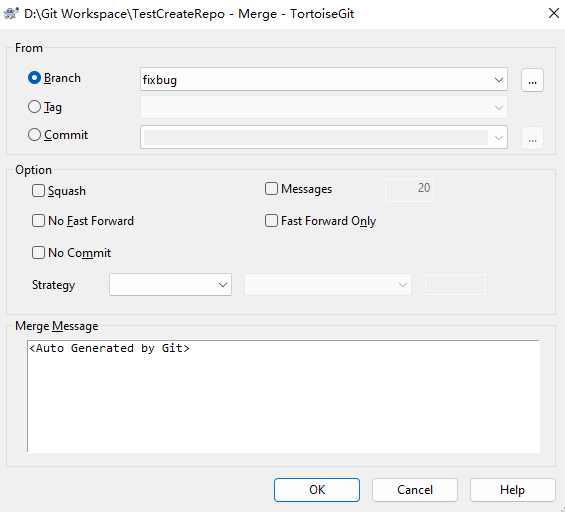
### 合并分支

假如我再fixbug分支修改了一个BUG并Commit了，我需要把这个分支合并到主分支master的话…

先切换到master分支



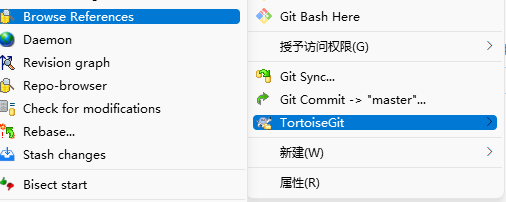
执行Merge From Branch选择fixbug

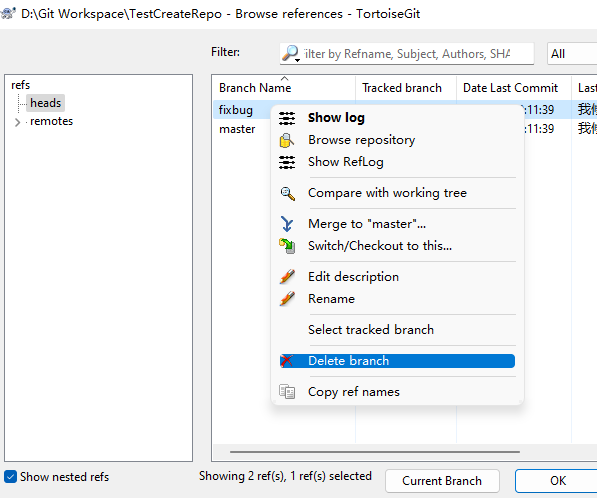


如果合并出现冲突，进入菜单项-Resolve…解决冲突

### 删除分支

在Browse References中



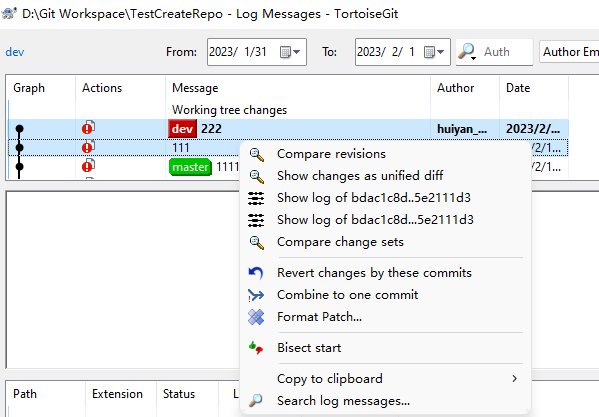


### 其他

#### 4.4.1 合并提交

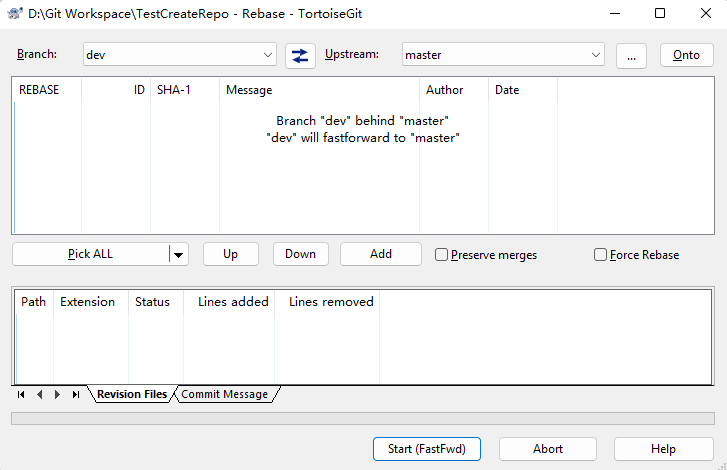
进入Show Log

假如我在Dev分支提交了2次，我可以选中2次提交点击Combine to one commit将他们合并为1次提交



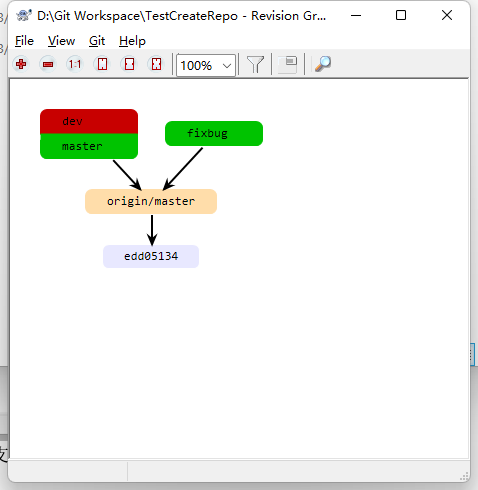
#### 4.4.2 Git Rebase

Rebase可以将DEV分支的基点移动到master分支最新节点。如下图设置Branch为dev，Upstream为master。



#### 4.4.3 分支图快捷操作

进入菜单项Revision Graph



红色：当前所在分支

绿色：本地分支

橙色：远程分支

直接右键图上的分支可以对分支进行切换/删除快捷操作