**Git操作手册**

# 目 录

[目 录 3](#_Toc483239423)

[1 概述 4](#_Toc483239424)

[1.1目的 4](#_Toc483239425)

[1.2适用范围 4](#_Toc483239426)

[1.3名词解释 4](#_Toc483239427)

[1.4参考资料 5](#_Toc483239428)

[2 操作手册 5](#_Toc483239429)

[2.1 Git客户端安装 5](#_Toc483239430)

[2.1.1 Git Bash安装 5](#_Toc483239431)

[2.1.2 TortoiseGit安装 6](#_Toc483239432)

[2.2 Git客户端配置 7](#_Toc483239433)

[2.2.1客户端公私钥生成 7](#_Toc483239434)

[2.2.2生成PPK文件 8](#_Toc483239435)

[2.2.3 TortoiseGit配置 12](#_Toc483239436)

[2.3 Gitlab端配置 13](#_Toc483239437)

[2.3.1 Gitlab端配置公钥 13](#_Toc483239438)

[2.4 克隆远程仓库 14](#_Toc483239439)

[2.5 下载/同步远程仓库到本地 16](#_Toc483239440)

[2.5.1 新建本地仓库 16](#_Toc483239441)

[2.5.2 下载远程仓库文件 16](#_Toc483239442)

[2.6 提交文件至本地仓库后推送至远程仓库 19](#_Toc483239443)

[2.7 推送本地仓库文件至远程仓库 23](#_Toc483239444)

[2.8 基于特定版本创建远程仓库的分支 26](#_Toc483239445)

[2.9基于特定版本创建本地仓库的分支 27](#_Toc483239446)

[2.10 本地仓库切换分支 28](#_Toc483239447)

[2.11 远程仓库推送冲突与解决 30](#_Toc483239448)

[2.12 合并分支和解决冲突 33](#_Toc483239449)

[2.13 当前版本与前一版的差异比较 35](#_Toc483239450)

[2.14 任意版本间的差异比较 38](#_Toc483239451)

[2.15 分配成员角色 41](#_Toc483239452)

[2.16 建立Group级别项目 42](#_Toc483239453)

[2.16.1 创建组 42](#_Toc483239454)

[2.16.2 创建Group级别项目 42](#_Toc483239455)

[2.17 设置受保护分支 43](#_Toc483239456)

[2.18 Fork远程仓库 44](#_Toc483239457)

[2.19 发起合并请求 45](#_Toc483239458)

[2.20 评审源代码并批准合并请求 47](#_Toc483239459)

# 1 概述

## 1.1目的

为了规范开发人员和配置管理人员在项目生命周期中使用Git的操作手法，在技术测试团队、和开发团队之间，对使用Git进行版本管理达成共识，特编写此文档，用于指导Git操作的手法。

## 1.2适用范围

本文档适用于开发团队和技术测试团队所有Git操作的手法。

## 1.3名词解释

**Fetch（获取）：**从远程仓库更新数据到本地仓库。注意 ：Fetch 只是将代码更新到本地仓库，你需要检出（check out）或与当前工作分支合并（merge）才能在你的工作目录中看到代码的改变。

**Pull（拉取）：**从远程仓库更新数据到本地仓库，并与当前工作分支合并，等同于 Fetch + Merge。

**Git clone（克隆）：**从服务端将项目的版本库克隆下来。

**Git init（初始化）：**在本地初始化仓库，在本地创建版本库的时候使用。

**Push（推送）：**将本地仓库中已提交（commit）的数据推送到指定的 remote，没有 commit 的数据，不会push。

**Master 分支：**主分支，所有提供给用户使用的正式版本，都在这个主分支上发布。

**Tags（标签）：**用来记录重要的版本历史，例如里程碑版本。

**Origin：**默认的 remote的名称。

**HEAD：**指向你正在工作中的本地分支的指针。

## 1.4参考资料

|  |  |  |
| --- | --- | --- |
| **序号** | **参考文件** | **备注** |
| 1 |  |  |
| ２ |  |  |

# 2 操作手册

## 2.1 Git客户端安装

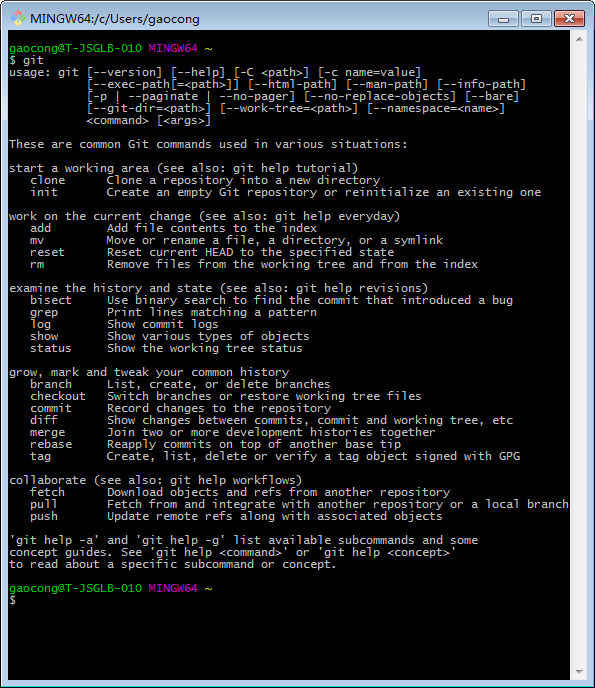
### 2.1.1 Git Bash安装

步骤1：下载Git293.zip，解压并默认步骤安装。

步骤2：安装完成后，启动Git Bash。



步骤3：键入“git”，如果出现以下信息，则安装成功。



### 2.1.2 TortoiseGit安装

步骤1：下载TortoiseGit-1.8.12.0-64bit.zip，解压并默认步骤安装。

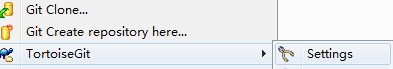
步骤2：安装完成后，右键查看，如弹出Git Clone菜单，表示成功。



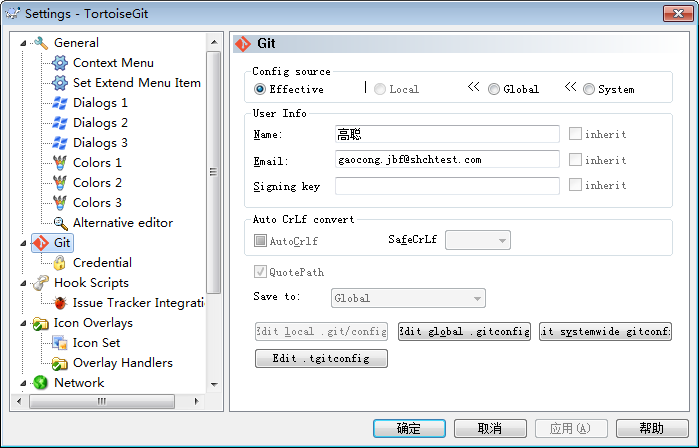
## 2.2 Git客户端配置

### 2.2.1 TortoiseGit配置

步骤1：右键，“TortoiseGit”，“Settings”。

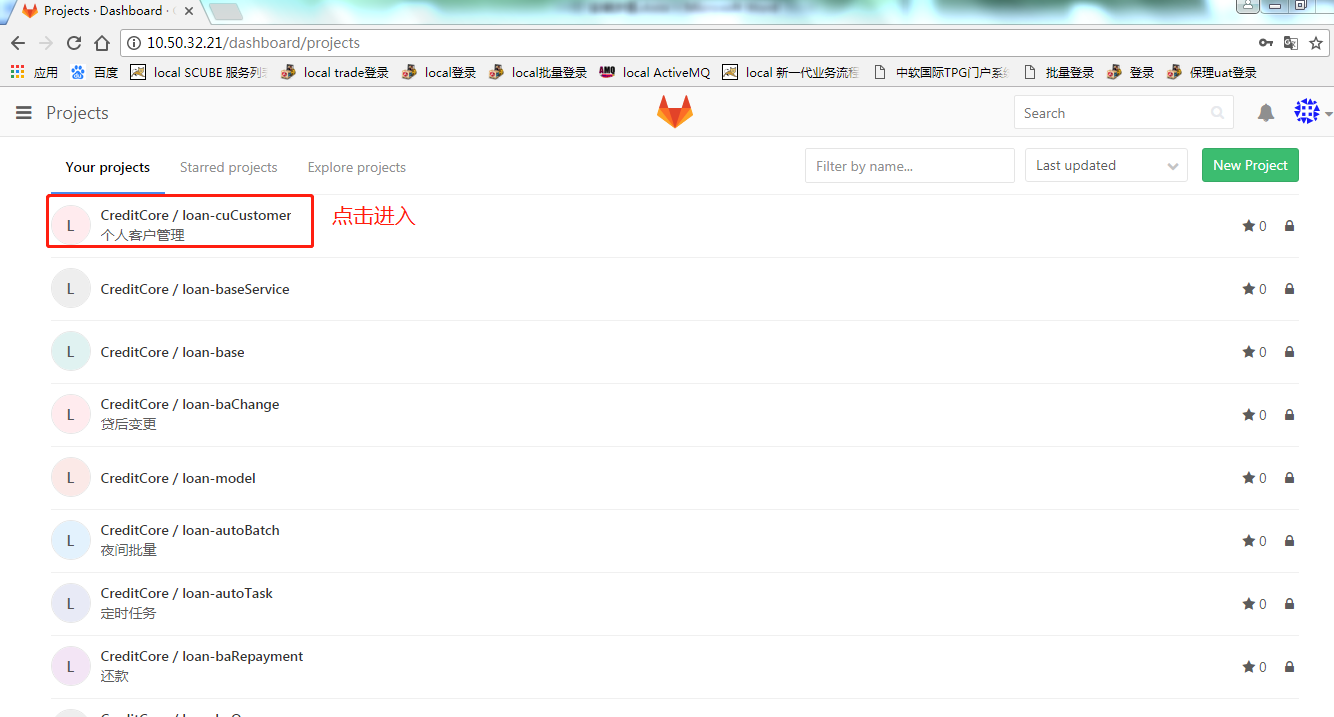


步骤2：配置“Name”和“Email”，确定；

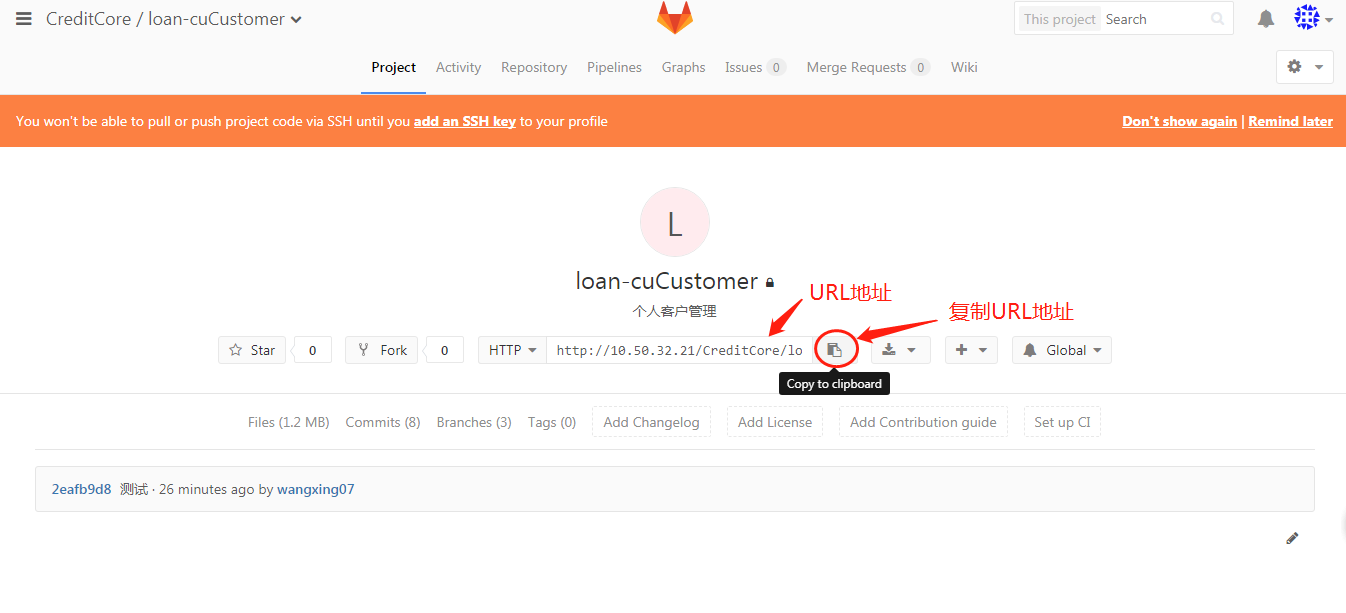


## 2.4 克隆远程仓库

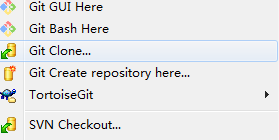
步骤1：访问登录远程仓库：<http://10.50.32.21/users/sign_in> 输入用户名密码

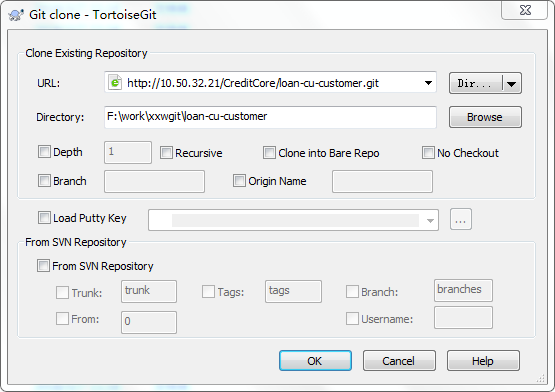


步骤2：复制URL

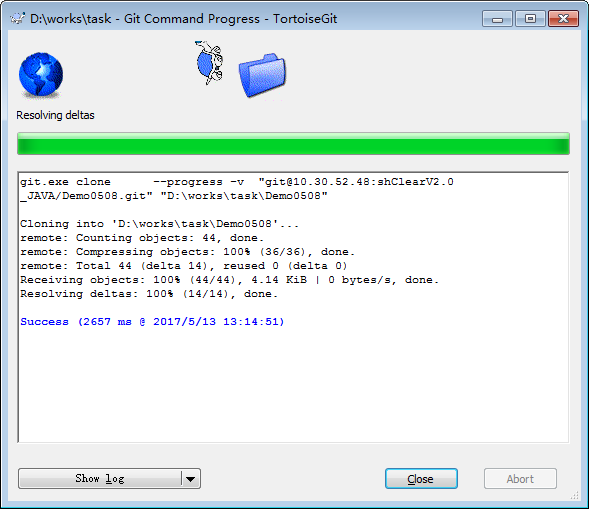


步骤3：在工作区中右键“Git Clone”，输入URL。



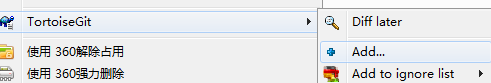


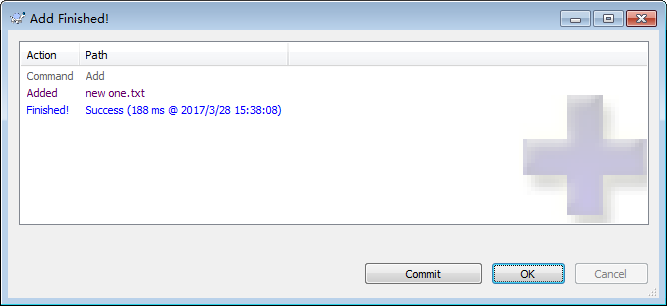
步骤2：ok。输入用户名密码，等待Git Clone完成。



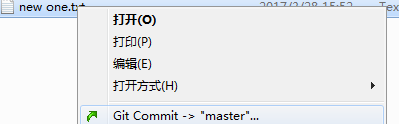
## 2.5 提交文件至本地仓库后推送至远程仓库

步骤1：选择要提交的文件，右键，“TortoiseGit”，选择“Add...”，并确定；

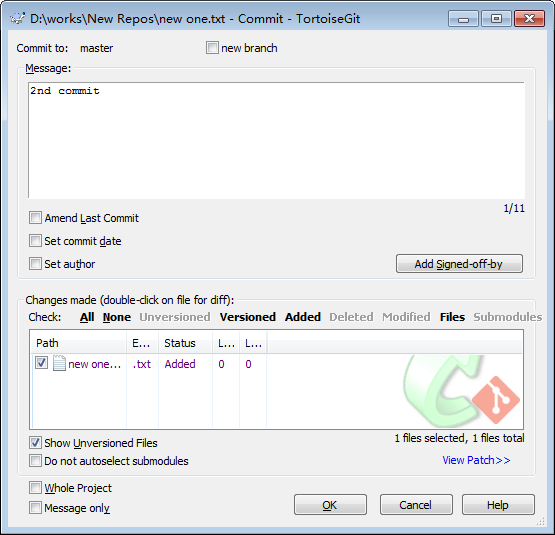




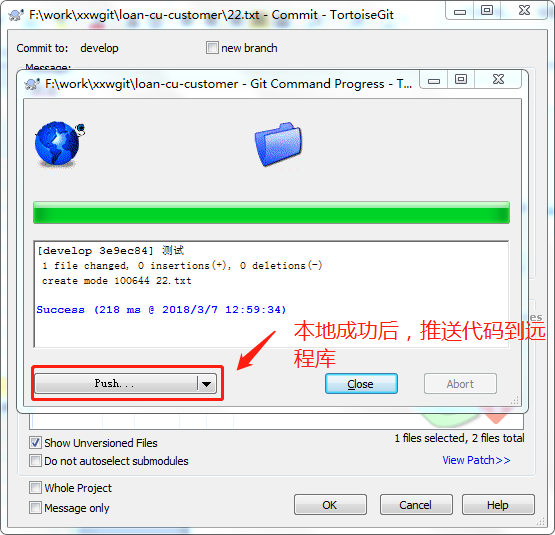
步骤2：选择要提交的文件，右键，“Git Commit -> "master"...”；

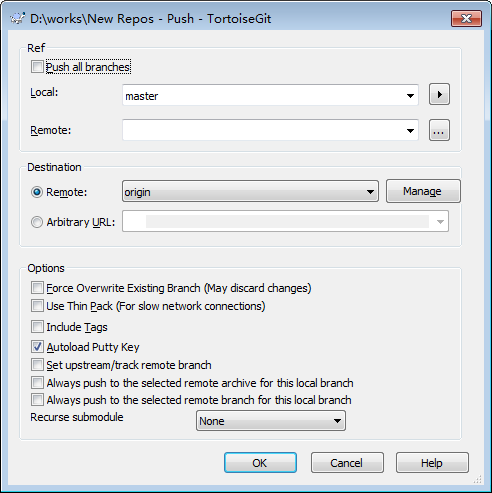


步骤3：填入提交的log信息，点击“OK”，直至上传结束。



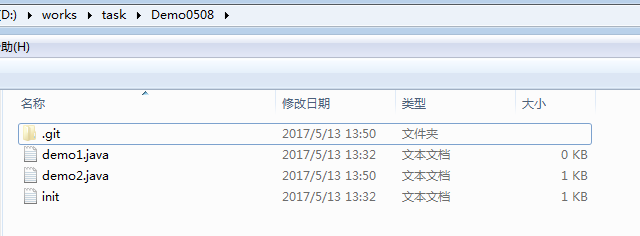
步骤4：步骤3结束后，点击“Push”，在弹出的对话框中点击“OK”。

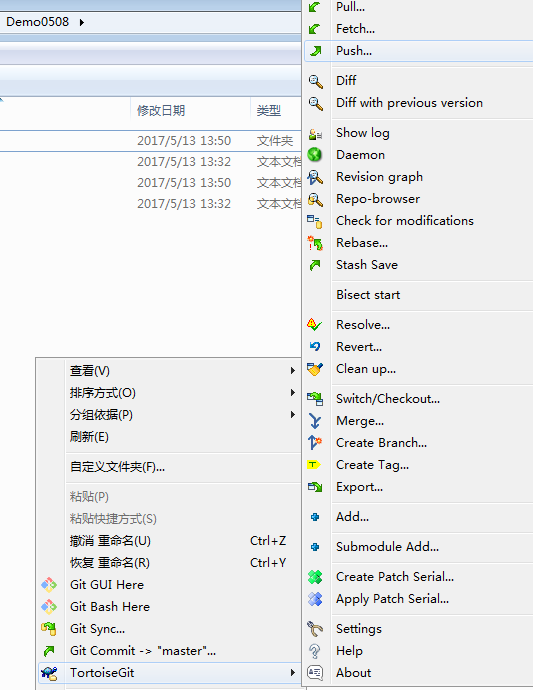




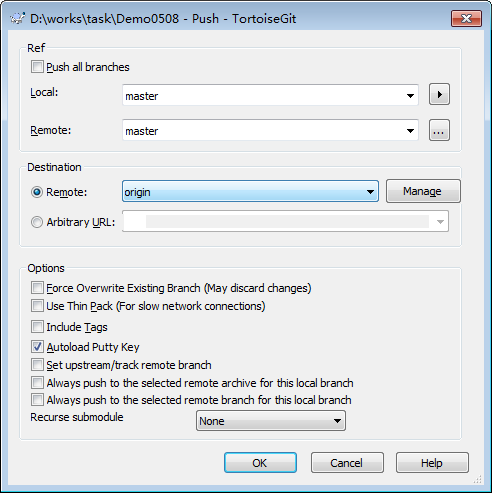
## 2.6 推送本地仓库文件至远程仓库

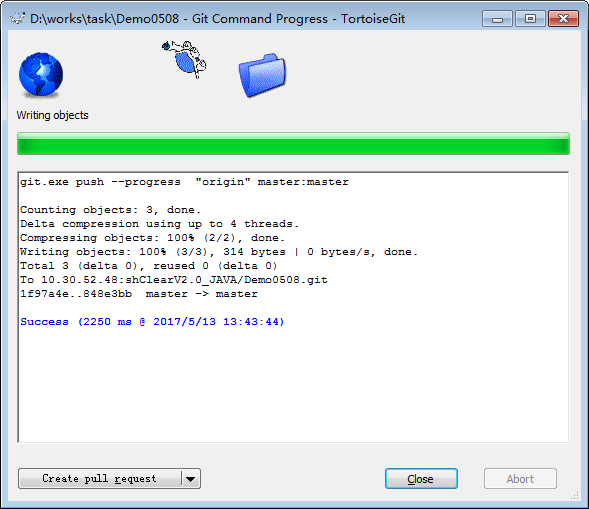
步骤1：在本地仓库右键，“TortoiseGit”，选择“Push”。





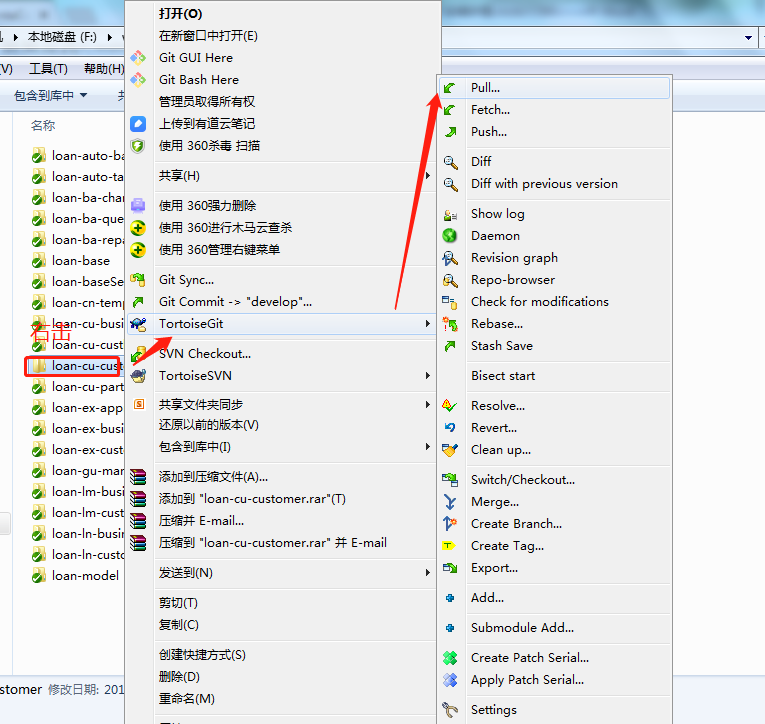
步骤2：选择远程仓库后，点击“OK”，等待Push完成。



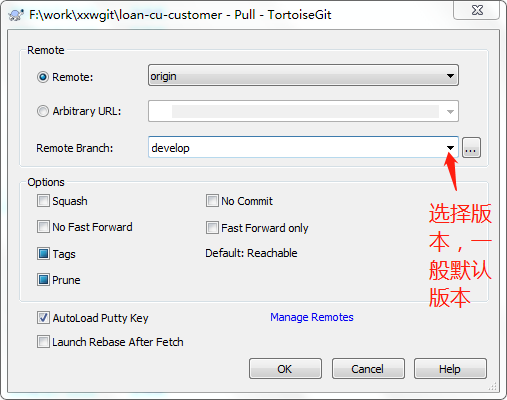


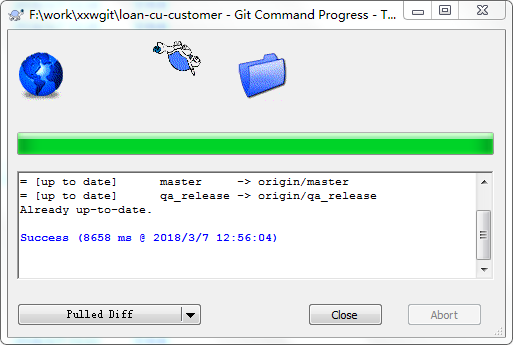
## 2.7 从远程仓库更新代码到本地

步骤1：选择本地仓库代码工程鼠标右击-》tortoiseGit-》pull



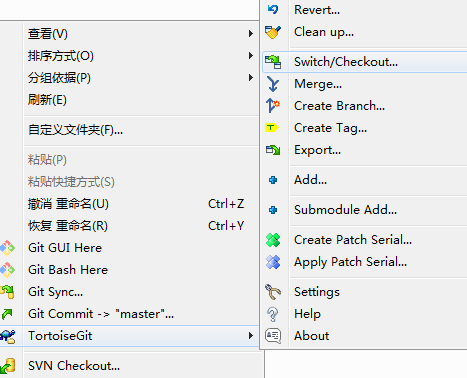
步骤2：选择版本（一般工程是多分支的，需要更新对应版本，如果不需要修改按照默认版本，点击ok输入用户名密码，直至更新完成即可）



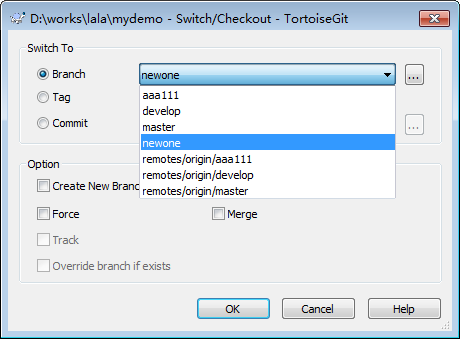


## 2.8 本地仓库切换分支

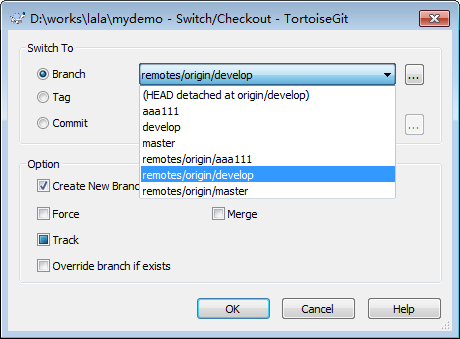
步骤1：本地仓库右键选择“TortoiseGit”，选择“Switch/Checkout..”。



步骤2：选择要切换到的分支“newone”，点击“OK”。

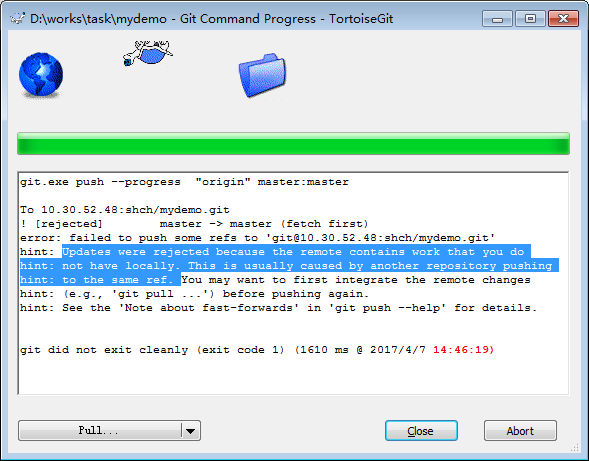


说明：该命令是可以基于任意Commit，已有分支，标签来实施切换。若是远程仓库分支，则默认在本地为该远程仓库分支创建一个新分支。如下图：

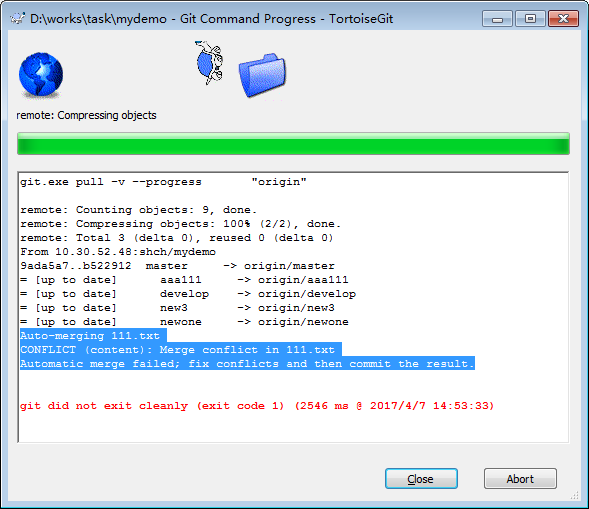


## 2.9 远程仓库推送冲突与解决

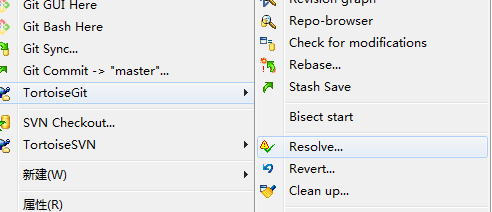
步骤1：在Git Push后产生冲突。

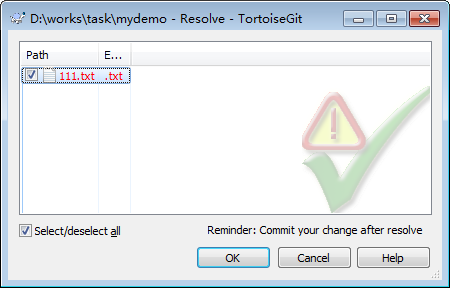


步骤2：Git Push冲突后，先Git Pull，会提示冲突信息。

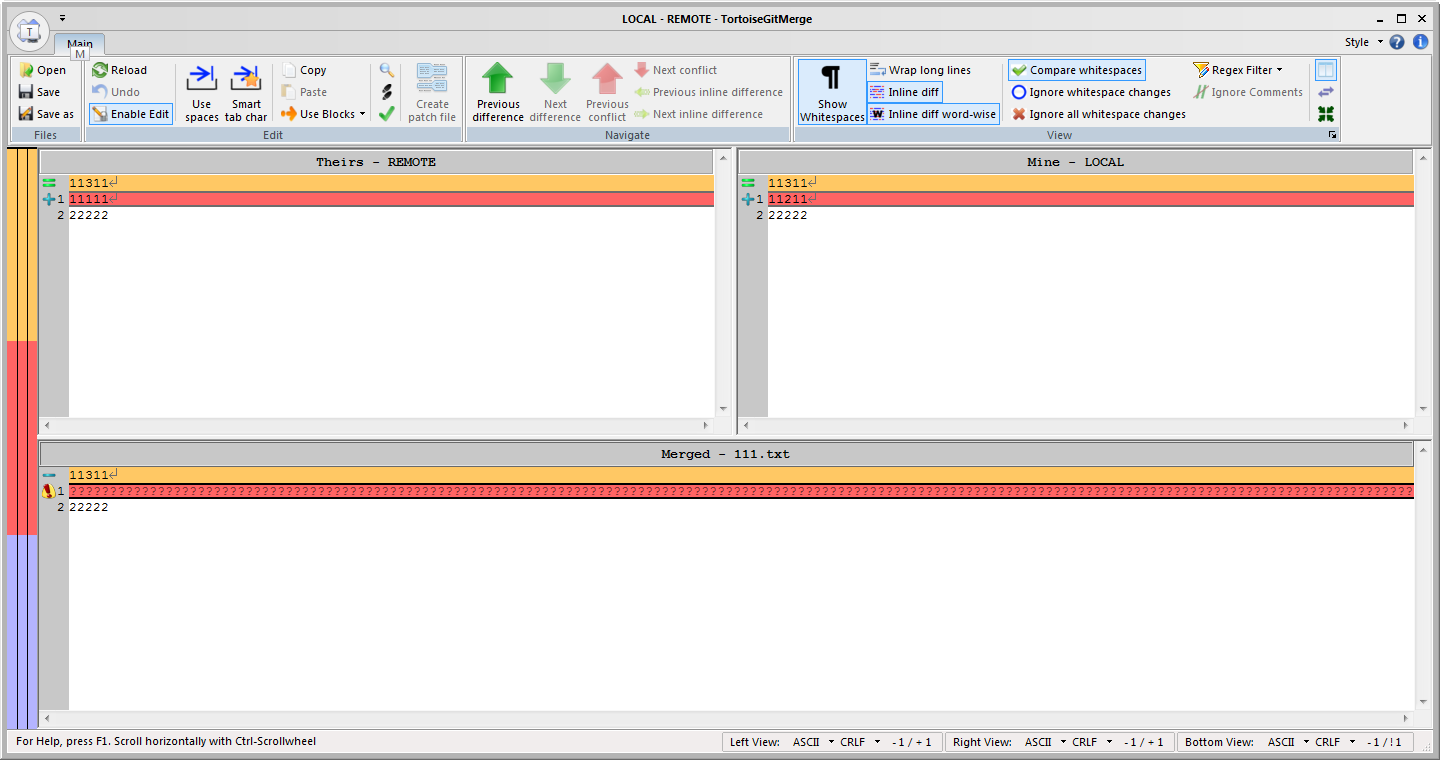


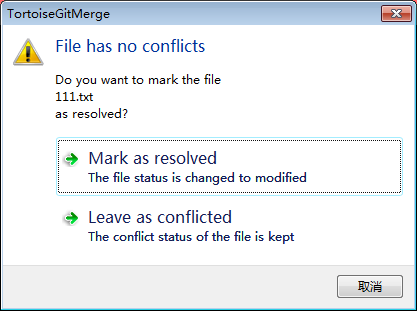
步骤3：右键“Resolve…”。





步骤4：双击冲突文件，在打开的编辑器中点击“Enable Edit”进行冲突解决，解决完成后点击“Save”，并标记冲突解决“Mark as resolved”；





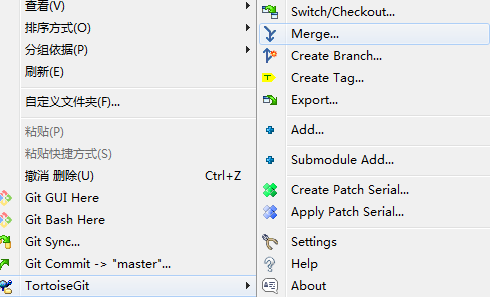
步骤5：最后执行Git commit，和Git push。

## 2.10 合并分支和解决冲突

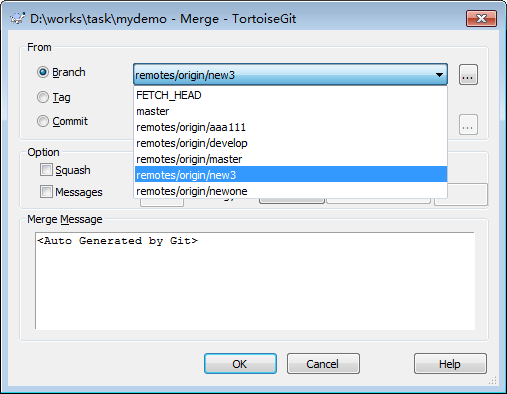
说明：如果想要将远程仓库A合并至远程仓库B，则需要先将B下载至本地，再将A合并至本地仓库B，最后提交并推送到远程仓库B。

步骤1：Git Pull或Git Clone远程仓库B到本地并切换到特定的分支。详细步骤请参考相应的操作手册。

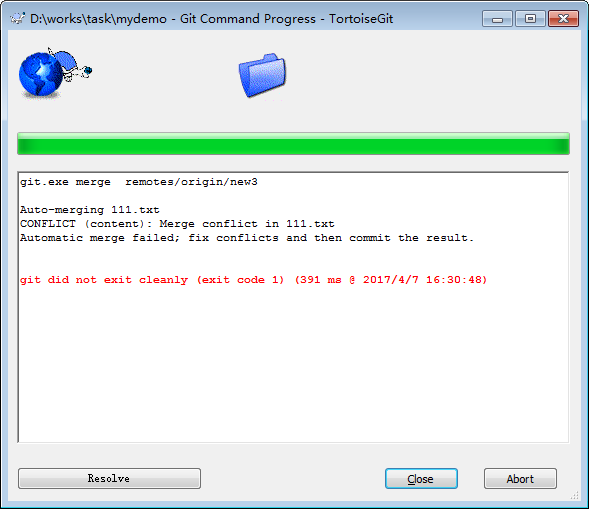
步骤2：在本地仓库右键选择“Merge…”。



步骤3：选择要合并到当前本地分支中的分支，例如：当前本地是“master”，选择“remotes/origin/new3”，则会将“remotes/origin/new3”的文件合并到“master”中；



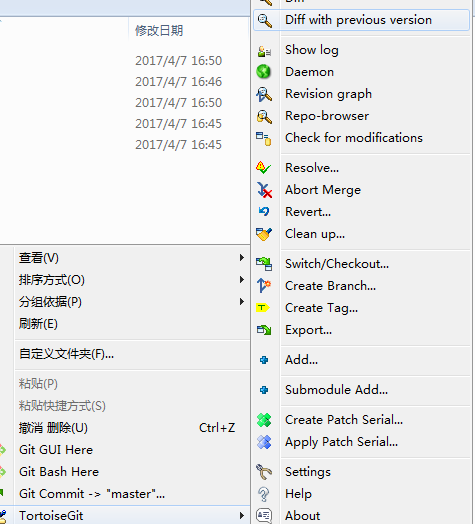
步骤4：若出现冲突，如下图，请按照2.11的步骤3和4解决冲突。



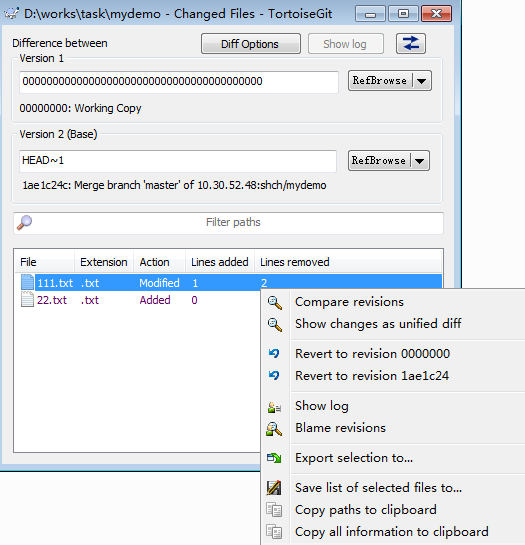
步骤5：最后执行Git commit，和Git push。

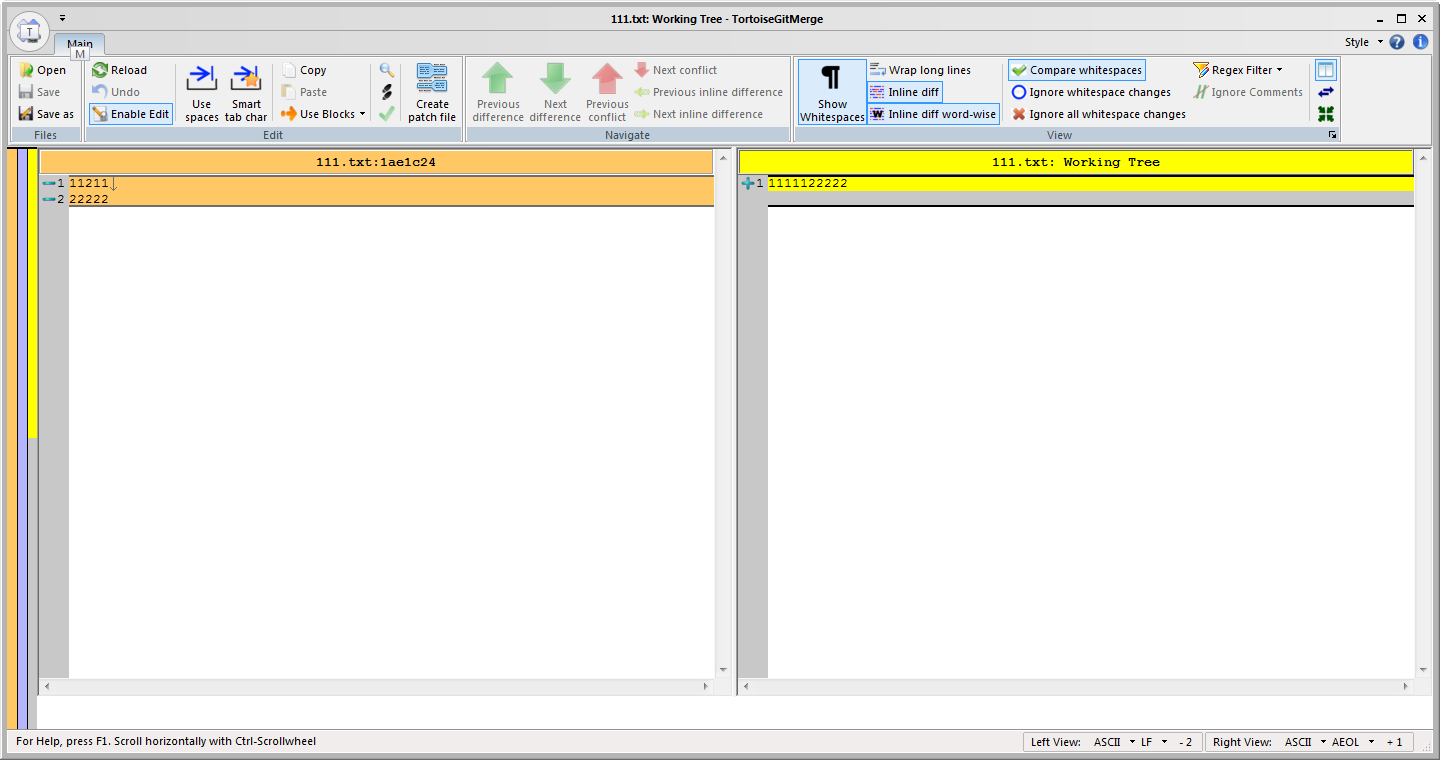
## 2.11 当前版本与前一版的差异比较

步骤1：右键“Diff with previous version”。



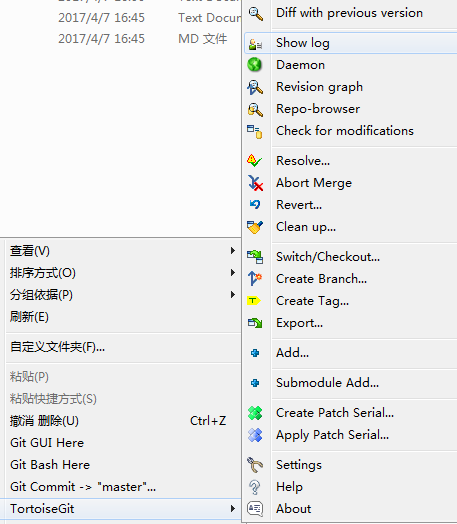
步骤2：选择文件，右键选择“Compare revisions”。



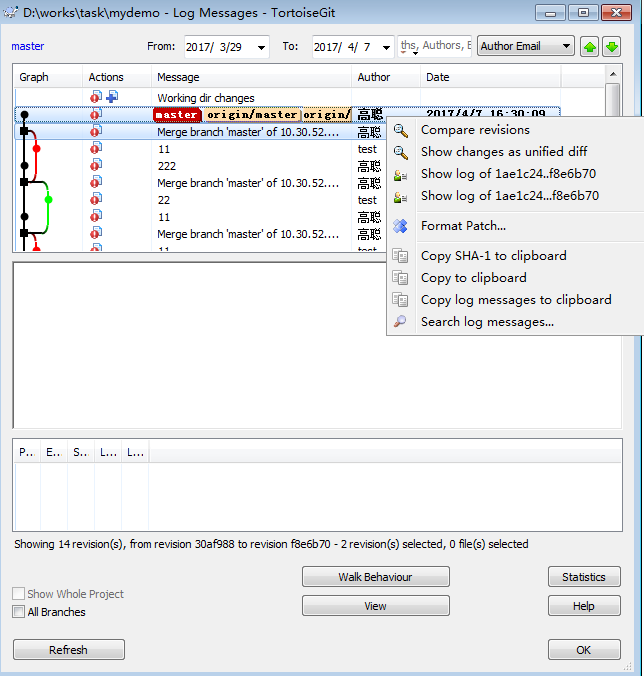


## 2.14 任意版本间的差异比较

步骤1：本地仓库右键，选择“Show log”。



步骤2：选中任意两个版本，右键选择“Compare revisions”。



步骤3：选择文件，右键选择“Compare revisions”

