Git简单使用手册

目录

[一、 软件安装 4](#_Toc527719056)

[二、 代码提交 4](#_Toc527719057)

[1. 记住用户和密码 4](#_Toc527719058)

[1) 方式一： 4](#_Toc527719059)

[2) 方式二： 4](#_Toc527719060)

[3) SSH方式： 4](#_Toc527719061)

[2. 克隆：git clone 5](#_Toc527719062)

[4) url的获取方式： 5](#_Toc527719063)

[5) Eclipse插件克隆方法： 6](#_Toc527719064)

[6) 命令方式： 11](#_Toc527719065)

[3. 分支切换：git checkout 11](#_Toc527719066)

[1) eclipse方式： 11](#_Toc527719067)

[2) 命令方式： 13](#_Toc527719068)

[4. 更新代码：Git pull 13](#_Toc527719069)

[1) Eclipse方式 14](#_Toc527719070)

[2) 命令方式： 15](#_Toc527719071)

[5. 提交至暂存区--git add 15](#_Toc527719072)

[1) Eclipse插件add方式： 16](#_Toc527719073)

[2) 命令add方式： 16](#_Toc527719074)

[6. 提交至本地库--git commit 17](#_Toc527719075)

[1) Eclipse插件commit方式： 17](#_Toc527719076)

[2) 命令方式： 17](#_Toc527719077)

[7. 提交至服务器--git push 17](#_Toc527719078)

[1) Eclipse插件push方式： 17](#_Toc527719079)

[2) 命令方式： 19](#_Toc527719080)

[三、 代码合并 20](#_Toc527719081)

[1. 没有冲突的合并： 20](#_Toc527719082)

[1) Eclipse合并方式： 20](#_Toc527719083)

[2) Stash操作： 22](#_Toc527719084)

[3) 命令方式： 24](#_Toc527719085)

[2. 存在冲突的合并解决 24](#_Toc527719086)

[1) Eclipse冲突合并方式： 24](#_Toc527719087)

[2) 命令方式： 26](#_Toc527719088)

[3) Stash合并操作： 28](#_Toc527719089)

[四、 其他常用命令： 29](#_Toc527719090)

[1. 还原修改： 29](#_Toc527719091)

[1) Eclipse方式： 29](#_Toc527719092)

[2) 命令方式： 30](#_Toc527719093)

[2. 分支创建： 31](#_Toc527719094)

[1) Jira操作： 31](#_Toc527719095)

[2) Stash操作： 32](#_Toc527719096)

[3) Eclipse方式： 34](#_Toc527719097)

[4) 命令方式： 34](#_Toc527719098)

[3. git status: 查看文件状态 35](#_Toc527719099)

[1) 命令方式： 35](#_Toc527719100)

[4. Git log：显示历史信息 35](#_Toc527719101)

[1) eclipse方式 35](#_Toc527719102)

[2) 命令方式： 36](#_Toc527719103)

[5. 分支查看 37](#_Toc527719104)

[1) 命令方式：git branch 37](#_Toc527719105)

# 软件安装

1. 首先安装：Git-2.19.1-64-bit

下载地址：

<https://git-scm.com/downloads>

1. 客户端：TortoiseGit 2.5.0.0(64bit) ，这个工具使用方式类似svn客户端。

# 代码提交

## 记住用户和密码

### 方式一：

安装了TortoiseGit则可以在本地文件.gitconfig中最后增加两行： [credential] helper = store，保存即可。

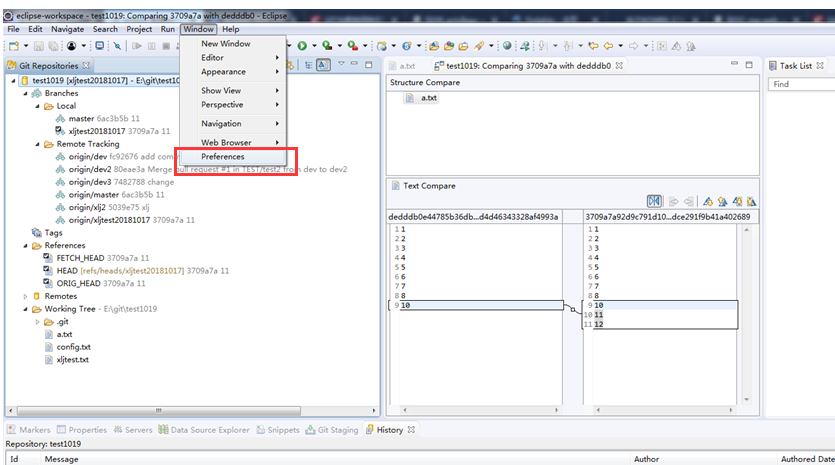
### 方式二：

在项目目录，右键->“git bash here”，执行下面命令：echo "[credential]" >> .git/config echo " helper = store" >> .git/config

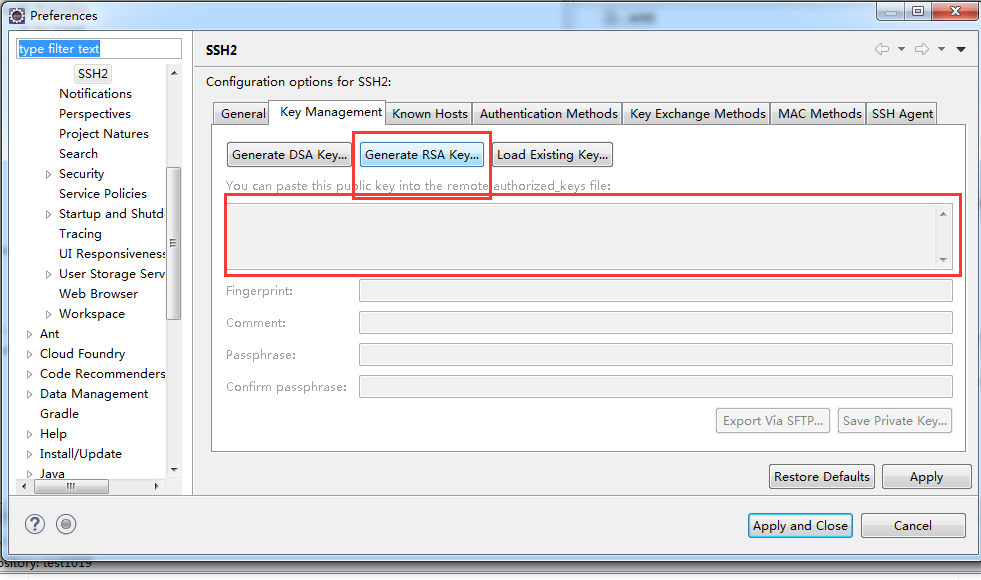
方式三：

### SSH方式：

Eclipse 配置git ssh 登录：



点击Key Management选项卡中的 generate RSA Key 即可生产秘钥，将生产的公钥在stash中添加即可。

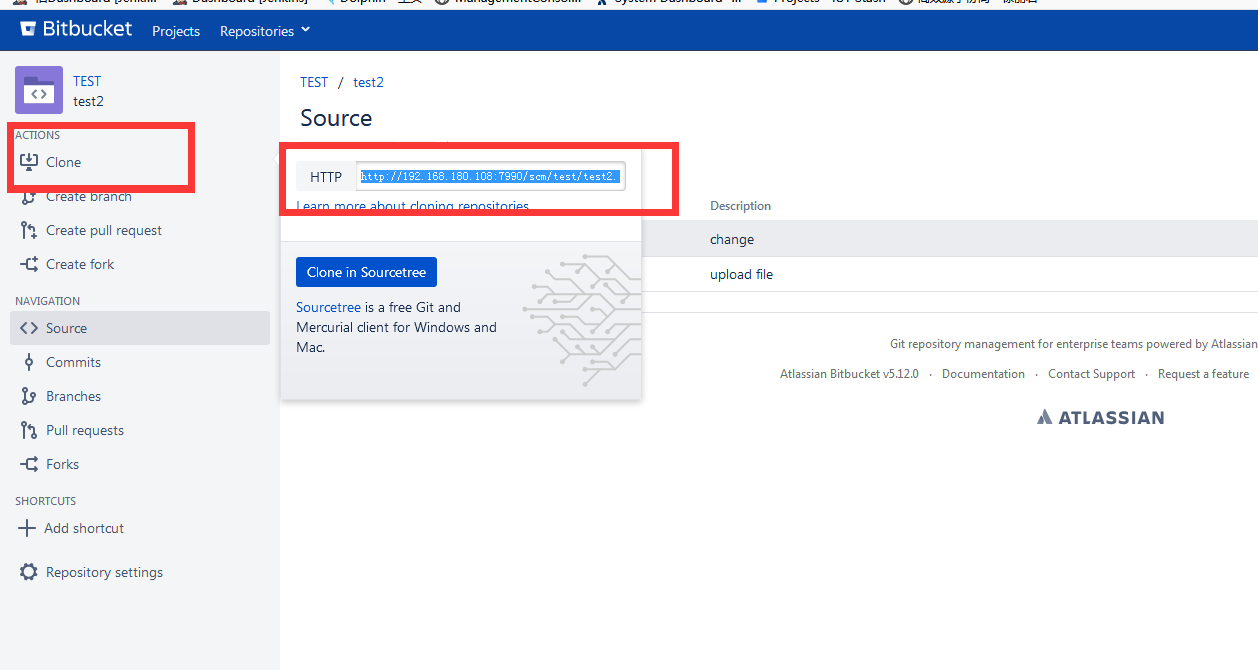


## 克隆：git clone

### url的获取方式：

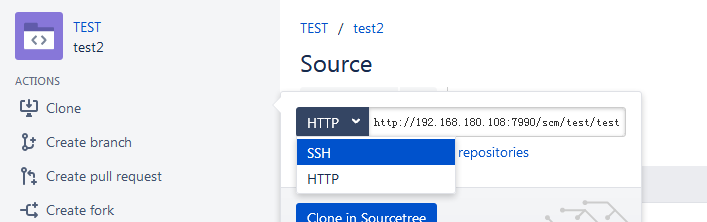
进入stash地址，选择对应库，获取url地址：

<http://192.168.180.108:7990/projects>



url分两一种，一种为http，一种为ssh

点击clone后，如下图可以选择所有需要的形式：

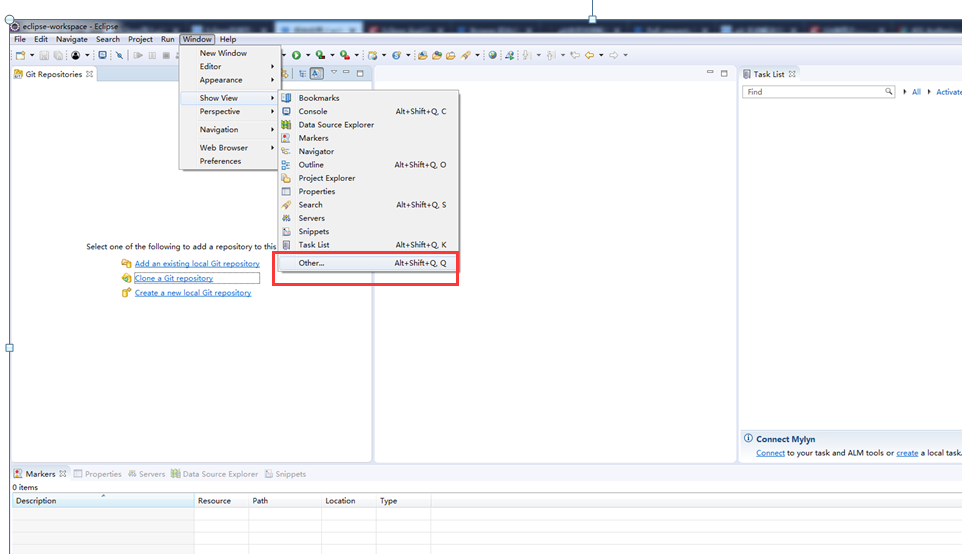


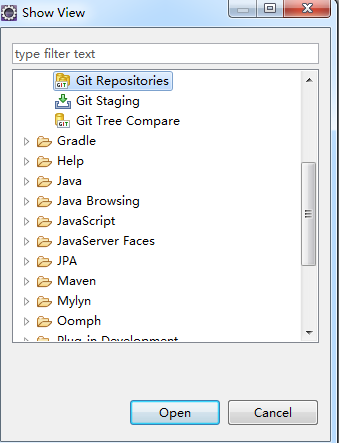
备注：在首次上测试环境/生产环境流程中，库地址需要填写ssh的形式，jenkins job都是ssh的方式拉去代码。

### Eclipse插件克隆方法：

1. 显示git仓库：

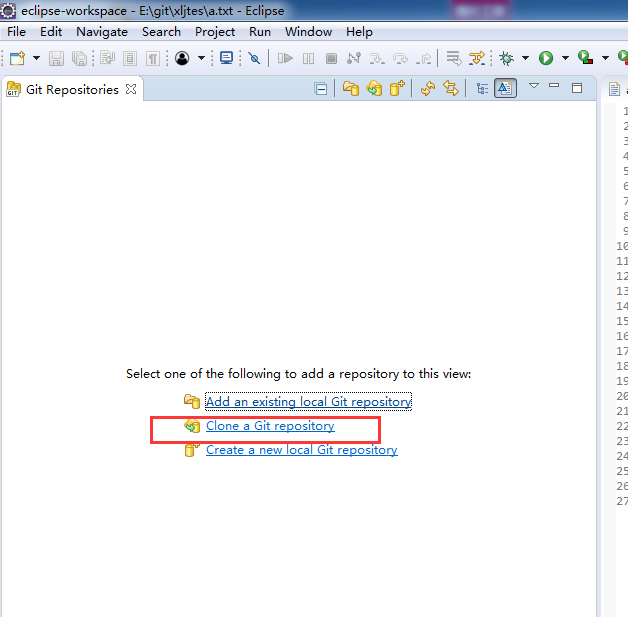
Eclipse中选择Windows->show view->other，然后选择git repo…即可显示git repo界面



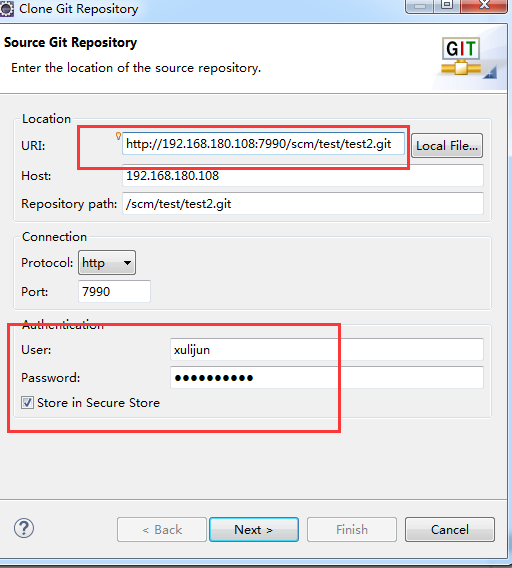


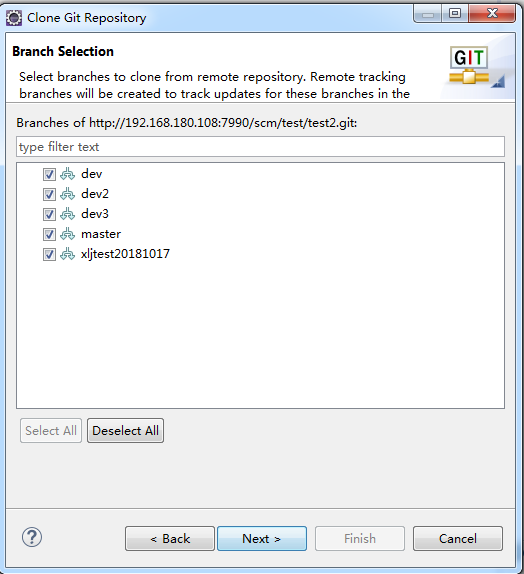
1. 克隆仓库（下载代码）

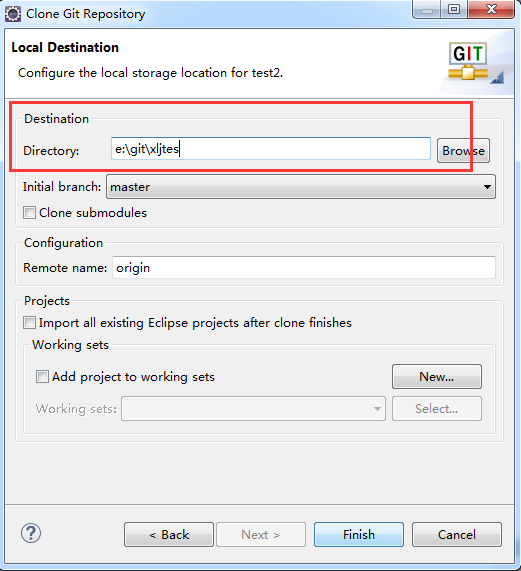
点击左侧clone a git repo…

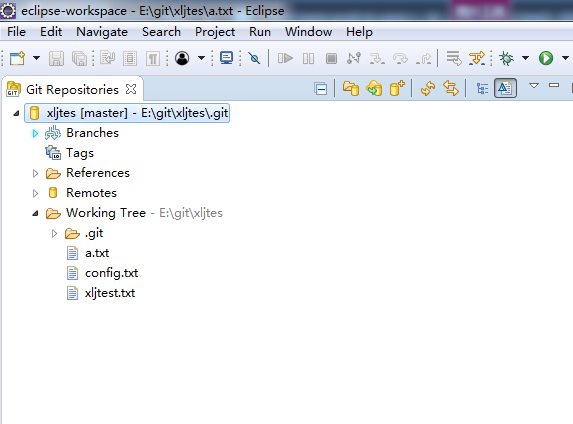


输入库的url地址，用户名和密码，即可显示库信息，按下图方式设置即可。





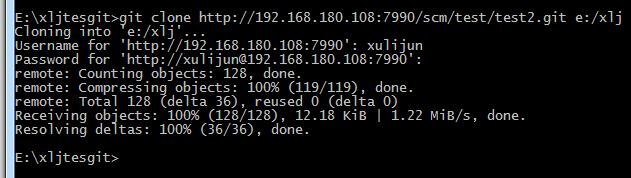




### 命令方式：

例如：克隆test2库至E盘xlj文件夹

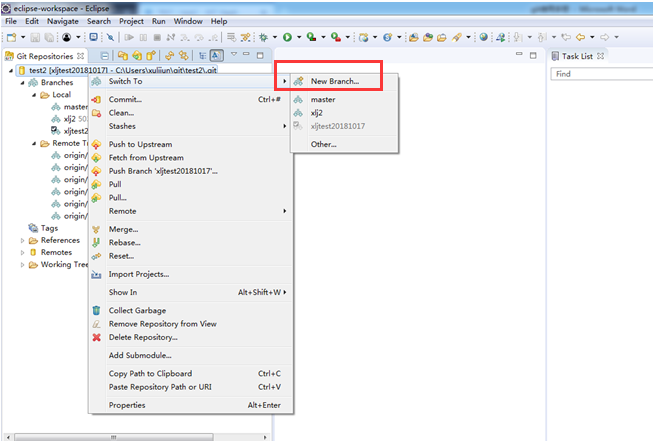
git clone <http://192.168.180.108:7990/scm/test/test2.git> e:/xlj

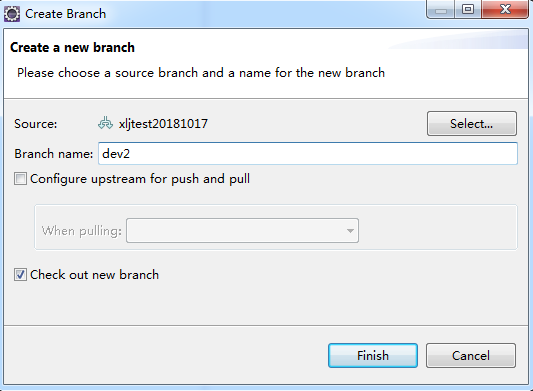


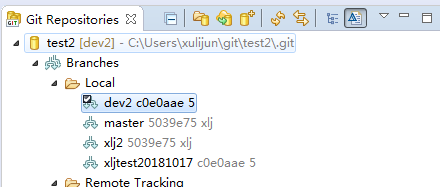
## 分支切换：git checkout

### eclipse方式：

直接在库名右键选择switch to new branch，输入分支名，即可完成切换，另外也可以直接点击左侧相应的本地和远程分支名直接切换分支。

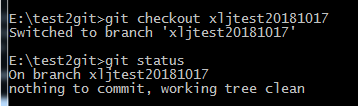






### 命令方式：

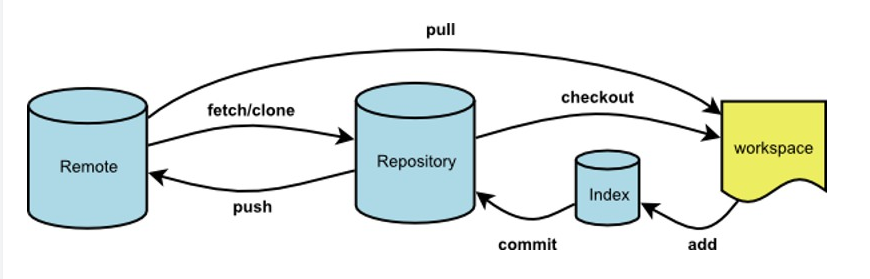
例如：git checkout xljtest20181017



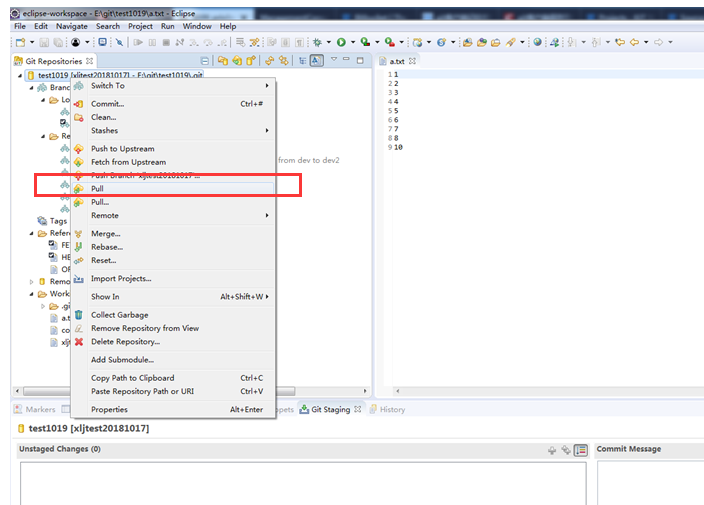
## 更新代码：Git pull

从远程服务器拉去最新代码，相当于svn的update，git pull=git fetch+git merge

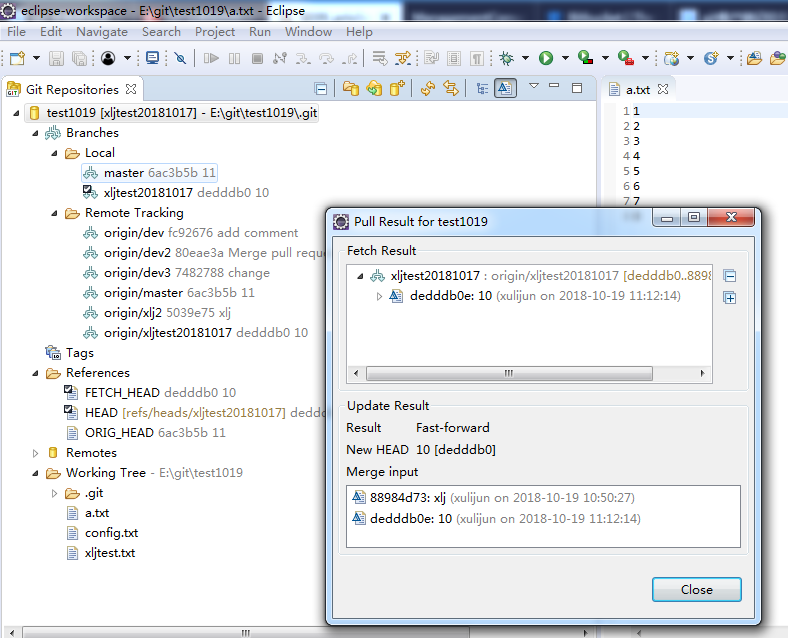
修改代码拉取分支时先git pull，以便于减少冲突。



### Eclipse方式

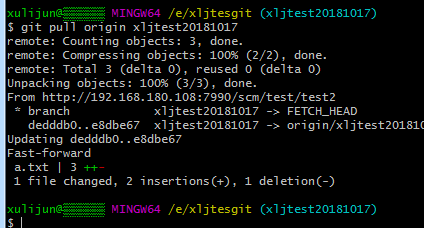


最后pull成功结果如下：



### 命令方式：

Git pull origin xljtest20181017：更新远程分支xljtest20181017至当前本地分支

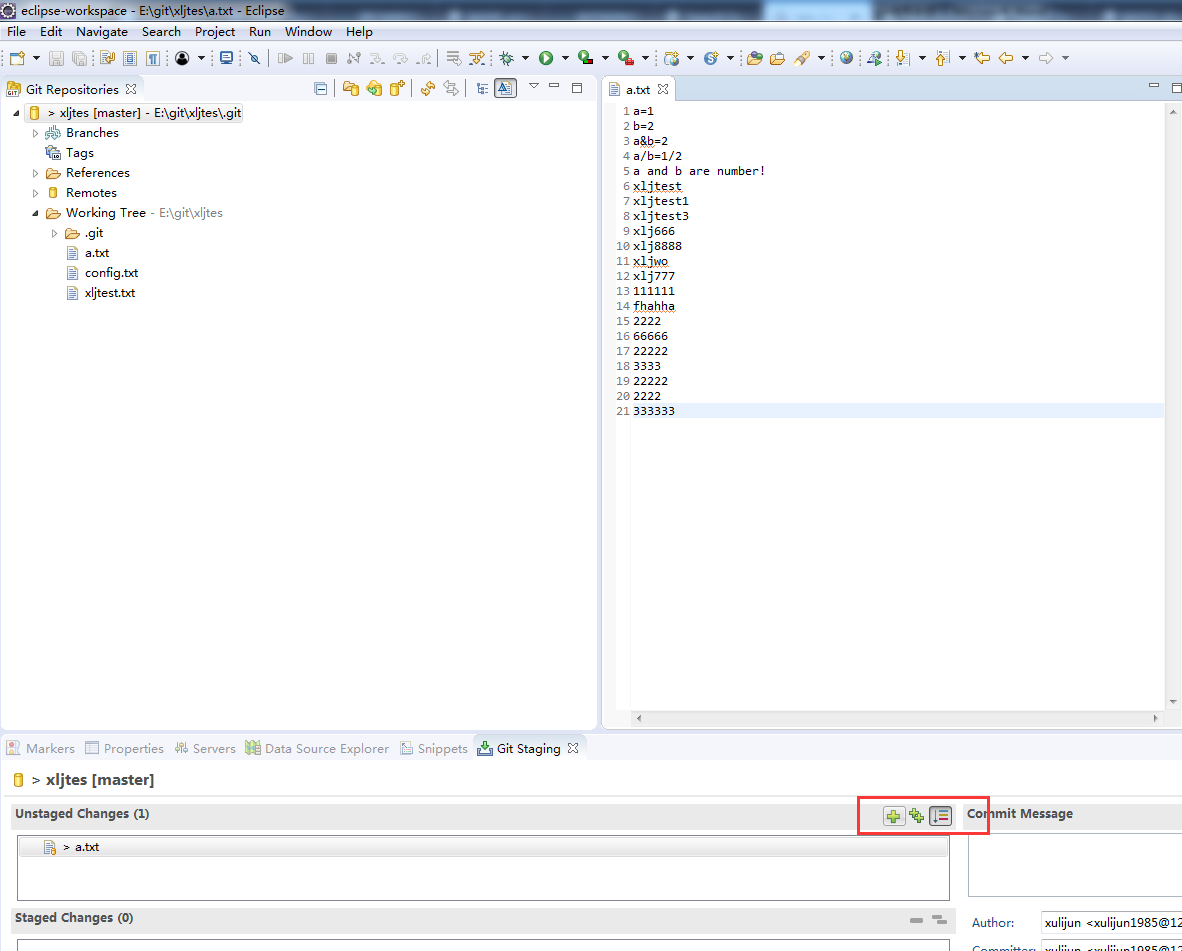


## 提交至暂存区--git add

在提交之前，Git有一个暂存区(staging area),可以放入新添加的文件或者加入新的改动. commit时提交的改动是上一次加入到staging area中的改动,而不是我们disk上的改动.会递归地添加当前工作目录中的所有文件。

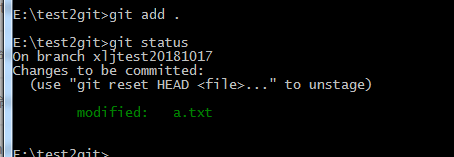
### Eclipse插件add方式：

在eclipse左下角会显示unstaged changes 、staged changes，commit message等区域，修改后的文件会在unstaged changes中，点击边上的+号即实现git add功能，如下图：



### 命令add方式：

例如：git add . ：将所有修改添加到暂存区

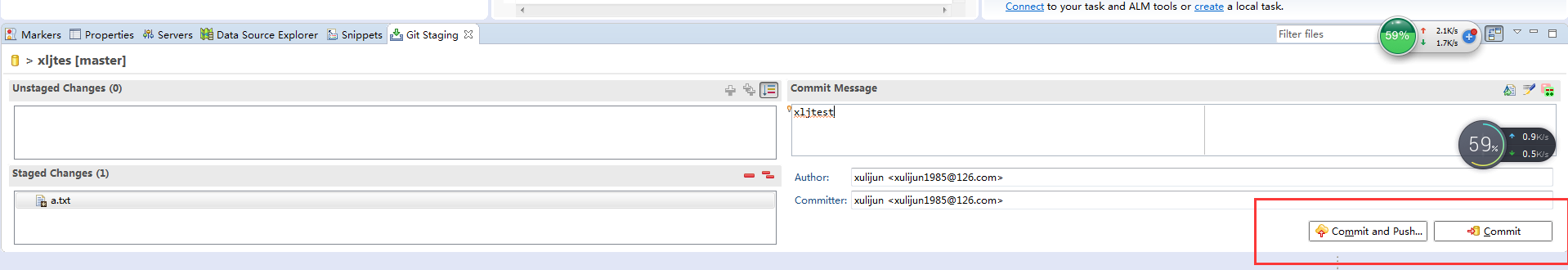


## 提交至本地库--git commit

提交已经被git add进来的改动，执行后提交至本地库中，并没有提交至服务器。

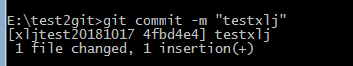
### Eclipse插件commit方式：

文件或者文件夹add后，输入修改日志，然后点击右下角commit按钮即可。



### 命令方式：

例如： git commit –m “xljtest”



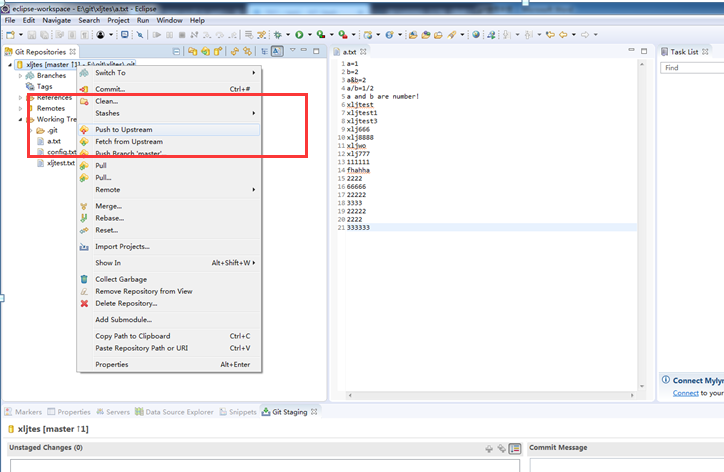
git commit -a ：会先把所有已经track的文件的改动`git add`进来，然后提交(有点像svn的一次提交,不用先暂存)。对于新增加的文件,还是需要执行`git add ` 命令。

## 提交至服务器--git push

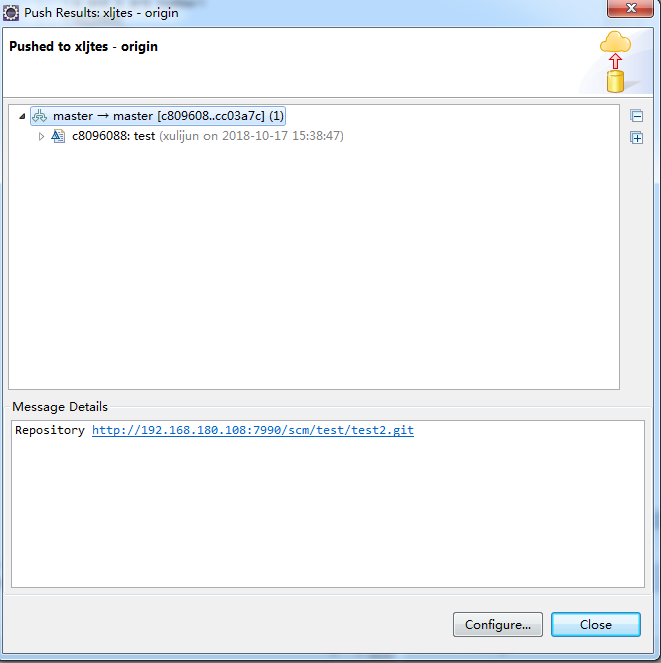
将加入暂存区的数据和快照提交至远程仓库。

### Eclipse插件push方式：

点击库名，然后点击push to Upstream按钮，即可。

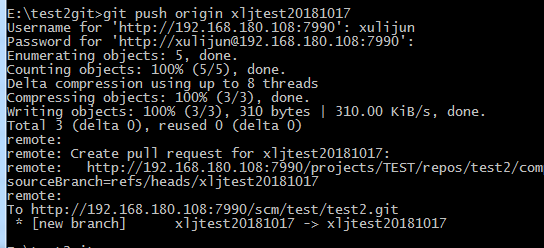


Push代码后显示如下，即将代码提交至远程服务器：



### 命令方式：

例如git push origin xljtest20181017

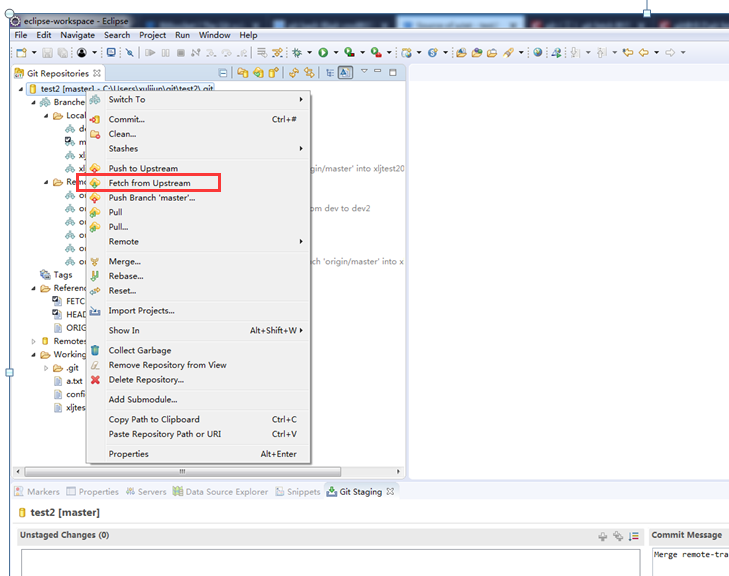


# 代码合并

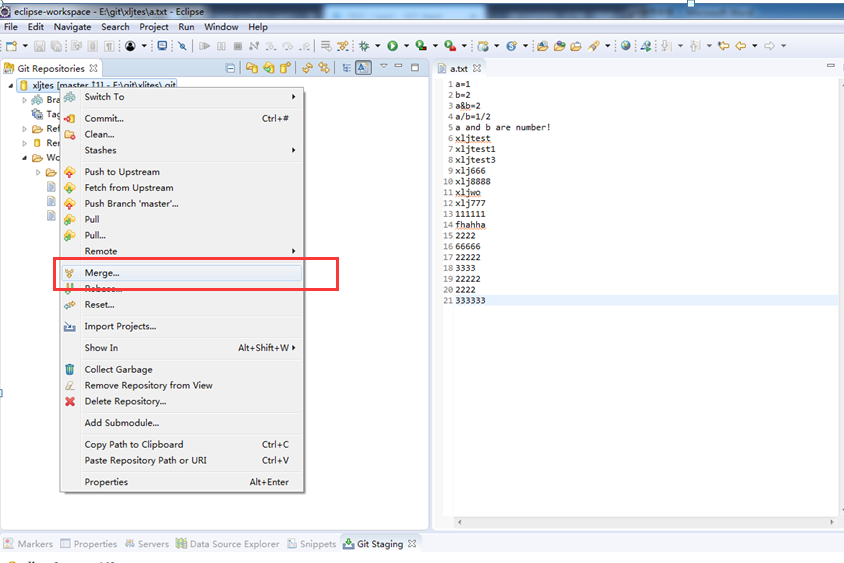
## 没有冲突的合并：

### Eclipse合并方式：

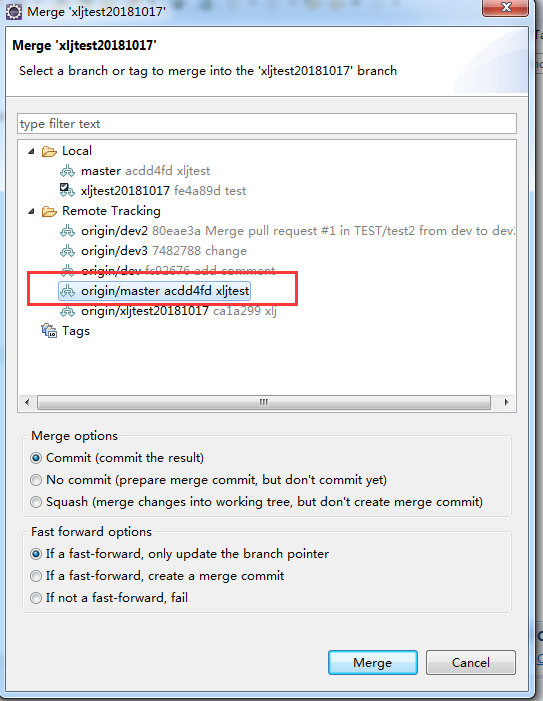
首先fetch from upstream ，相当于从远程获取最新版本到本地，但不自动合并。



然后库名右键选择merge

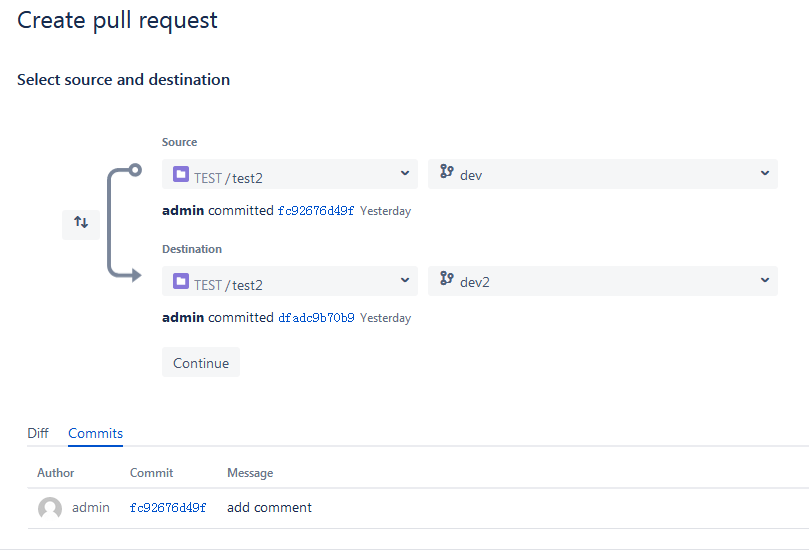


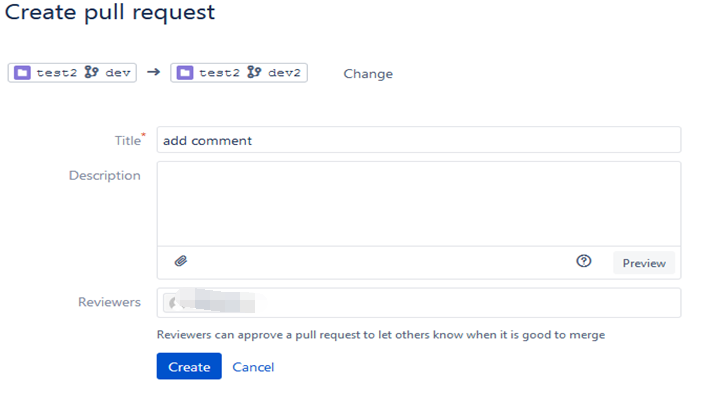
本地为xljtest20181017分支，这进行如下操作后，即将远程master分支合并到xljtest2017分支上，默认情况下执行以下操作已进行commit操作，需要再执行push后才能将内容提交至服务器。

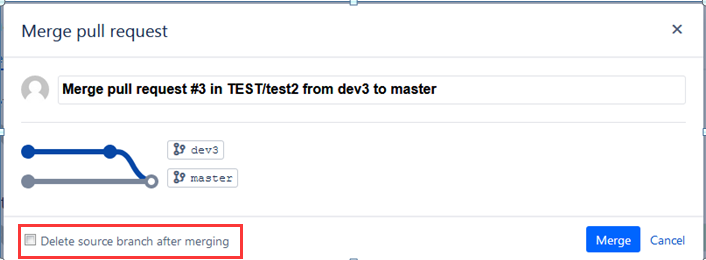


### Stash操作：

1.创建pr，即源分支->目的分支 的合并请求：如需要将dev分支合并到dev2分支，则如下操作：







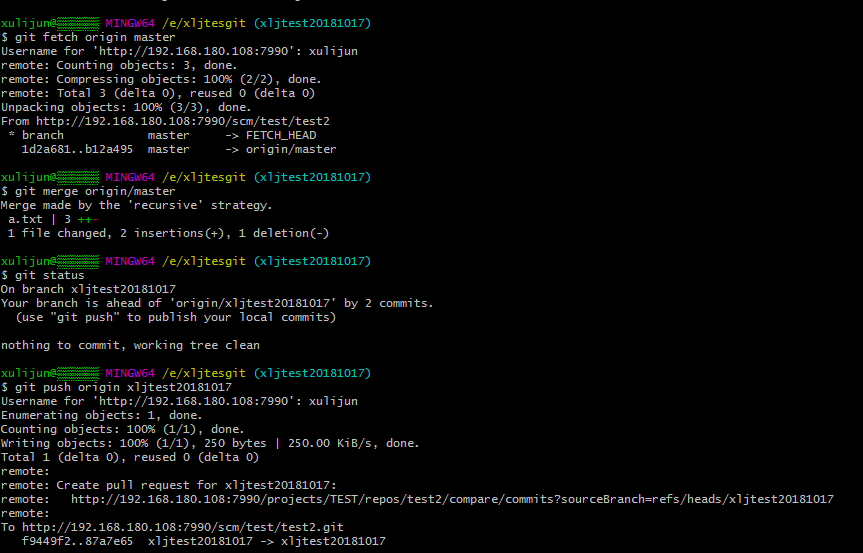
### 命令方式：

例如：git fetch origin master

git merge origin/master

git push origin xljtest20181017

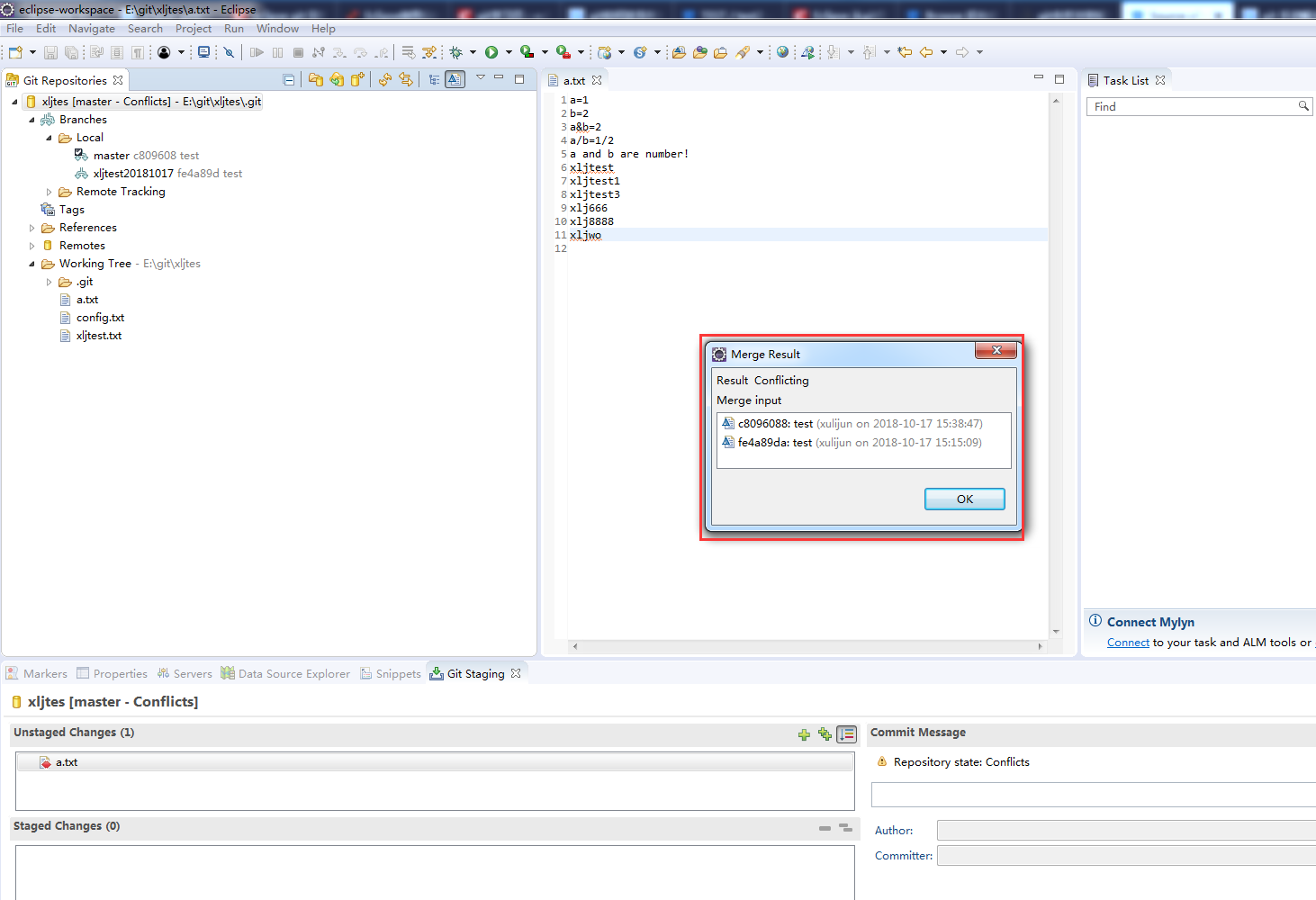
当前分支为xljtest20181017，即将远程master合并到xljtest20181017分支上。



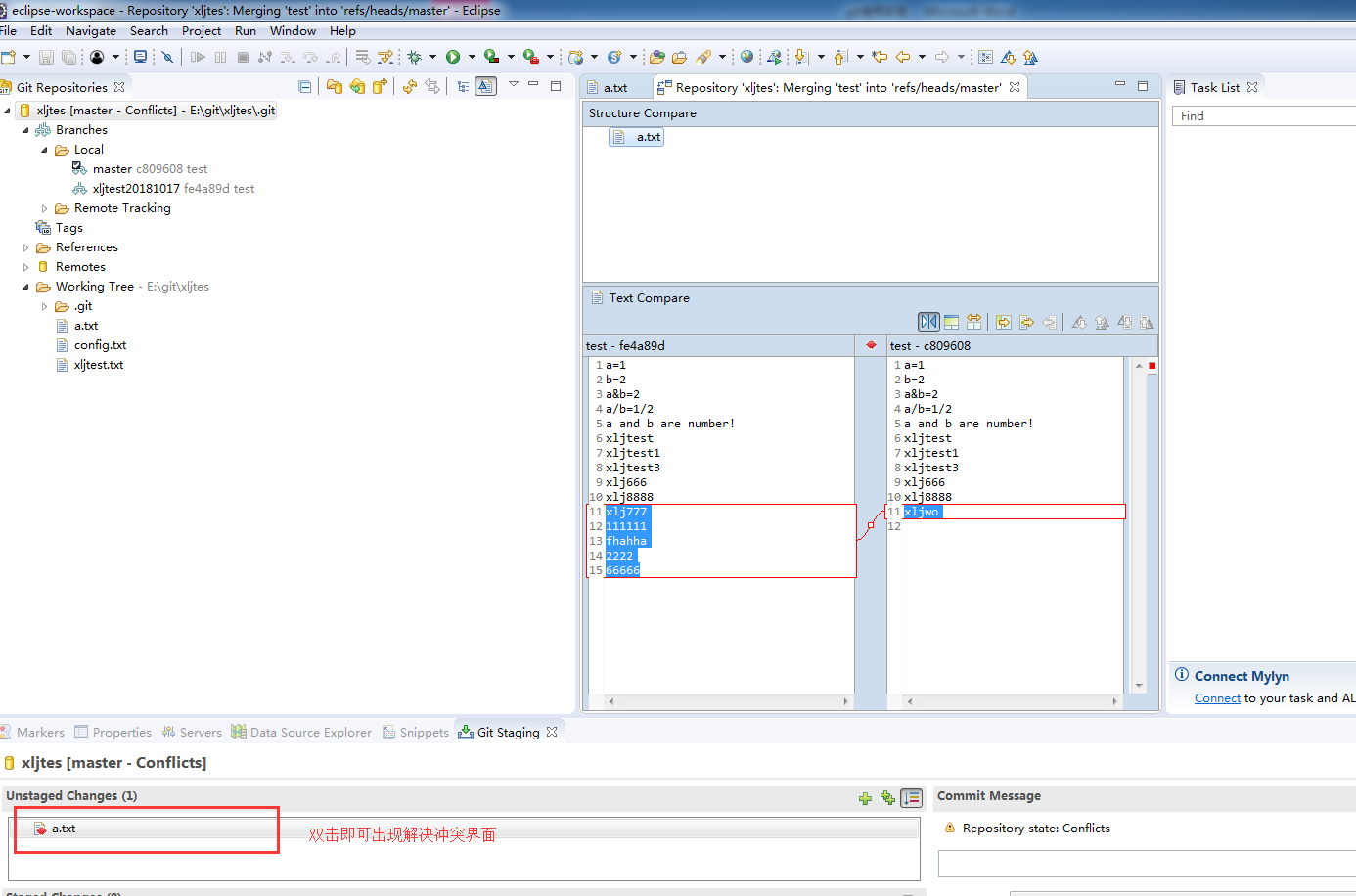
## 存在冲突的合并解决

### Eclipse冲突合并方式：

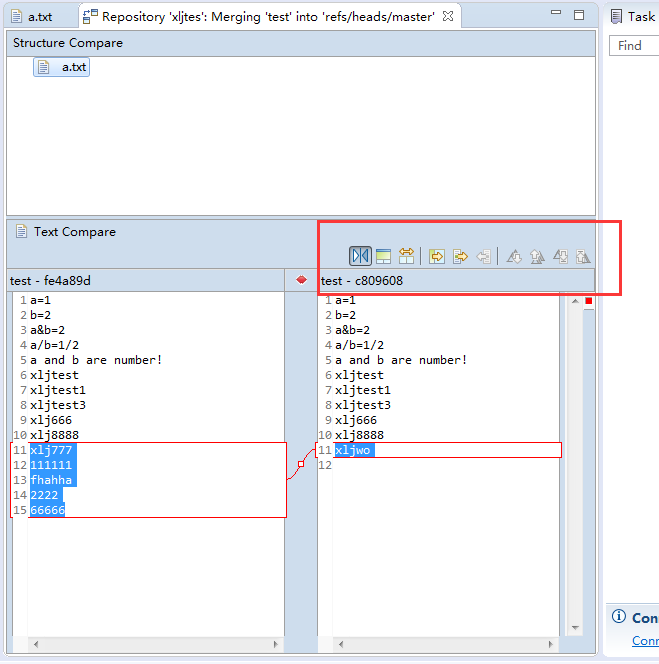
当合并存在冲突时会出现如下界面：



点击左下角冲突文件会具体显示冲突内容：



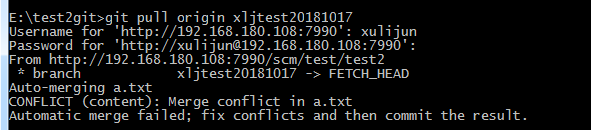
冲突解决界面提供一些工具，可按需选择解决冲突，如下图：



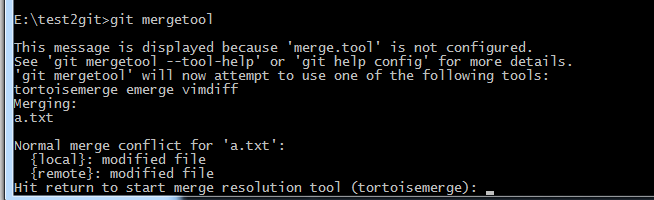
冲突解决后，git add->commit and push即可提交至服务器。

### 命令方式：

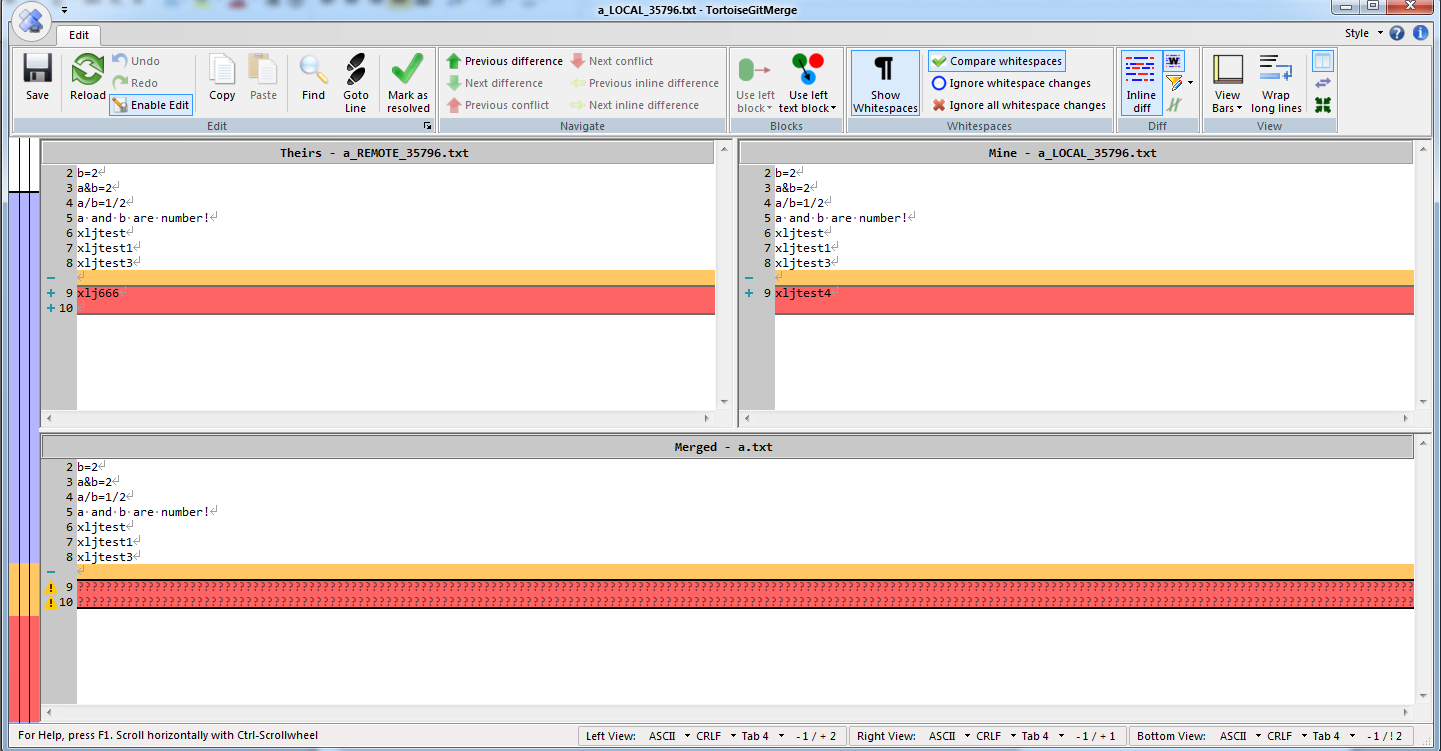
当前分支为master，执行git merge xljtest20181017/git pull origin xljtest20181017时，出现冲突。



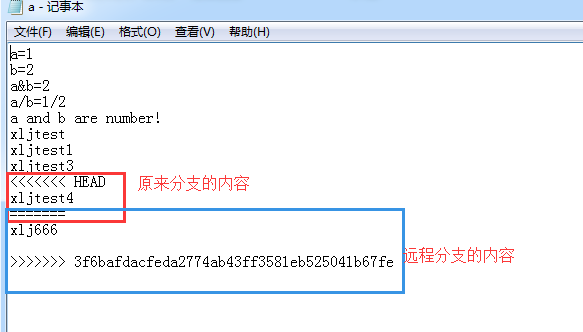
执行git mergetool 命令：



按enter键，出现合并界面，按所需选择合并即可。



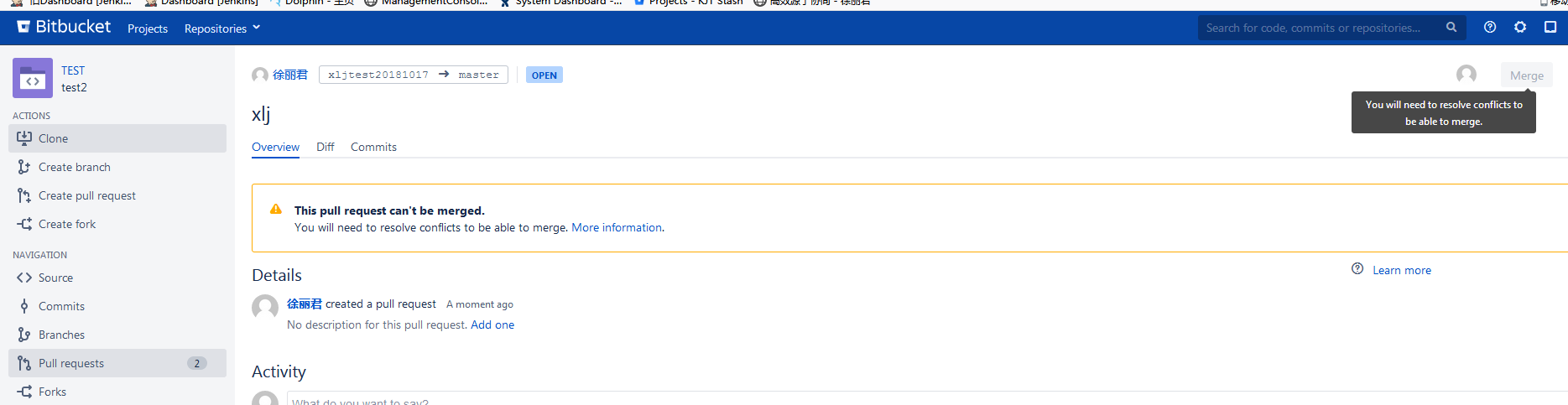
手动解决冲突方式如下图，按需要选择的内容即可。



### Stash合并操作：

提pr（合并请求是会出现提示，有冲突需要解决）。

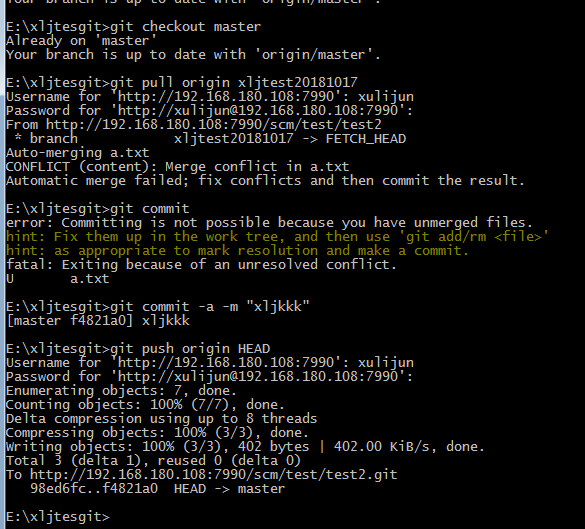
备注：如review者合并，那么合并完后stash上pull requests会自动消失，如授权给其他人合并，那么还需要去stash的pull requests中点下merge。



点击右上角提示，可操作方法，根据提示操作即可，冲突解决方法可按之前介绍的方式。



实际操作如下：

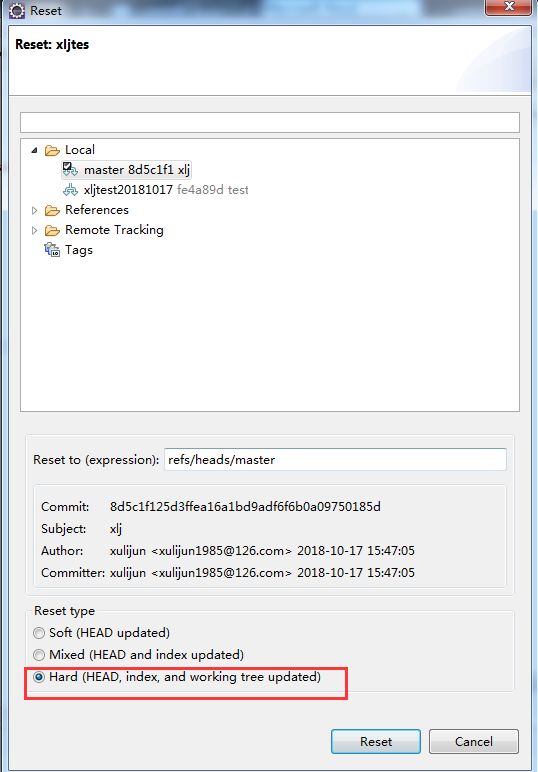


# 其他常用命令：

## 还原修改：

### Eclipse方式：

库名右键，选择reset：

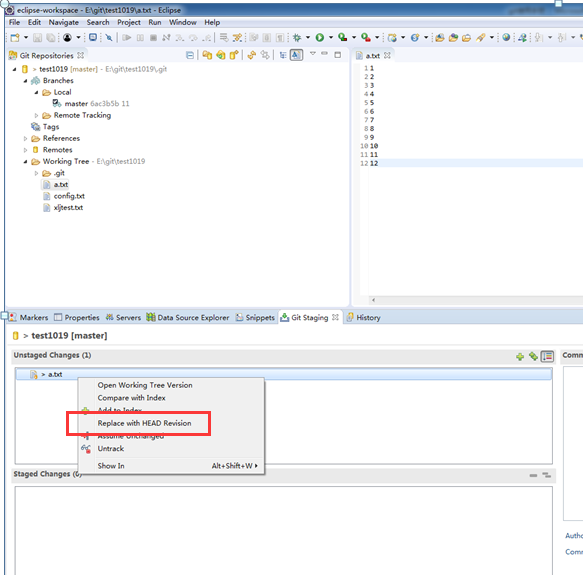


Hard 选项表示：彻底回退到某个版本，本地的源码也会变为上一个版本的内容。

Mixed选项表示：它回退到某个版本，只保留源码，回退commit和index信息。

Soft：回退到某个版本，只回退了commit的信息，不会恢复到index file一级。如果还要提交，直接commit即可。

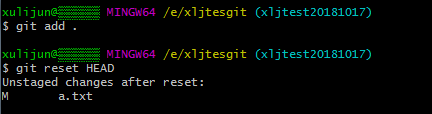
Eclipse在git staging窗口也可以使用右键还原文件的修改：



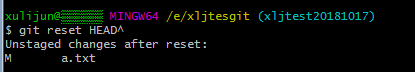
### 命令方式：

Git reset

如：git reset HEAD ：回退所有内容到上一个本地库版本(HEAD回退add的版本)



Git reset HEAD^：　回退到上一个提交的版本。



Git reset 057d：回退到057d版本。

Git reset –hard origin/master　将本地的状态回退到和远程的一样。

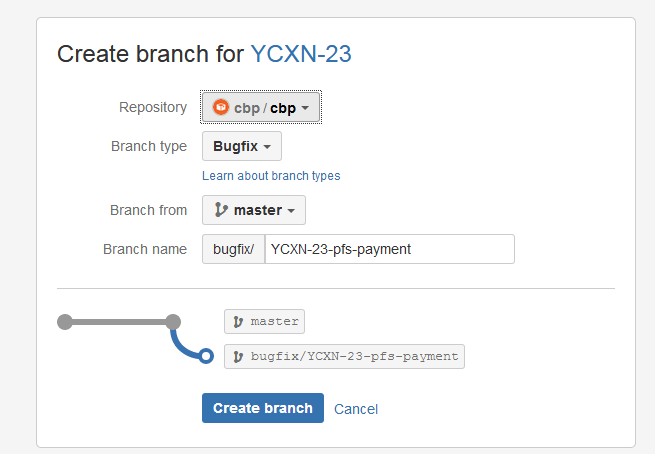
## 分支创建：

### Jira操作：

打开需求jira，点击右侧create branch：

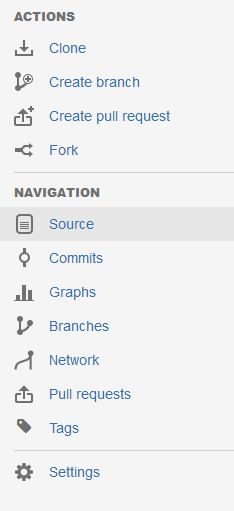


点击create branch后出现拉分支界面，按需求创建分支即可：

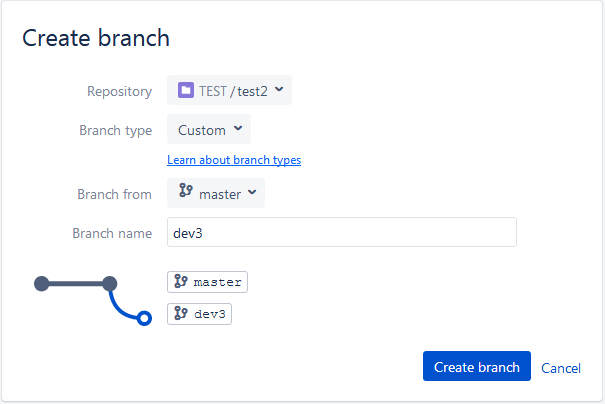


### Stash操作：

进入stash，然后点击creat branch：

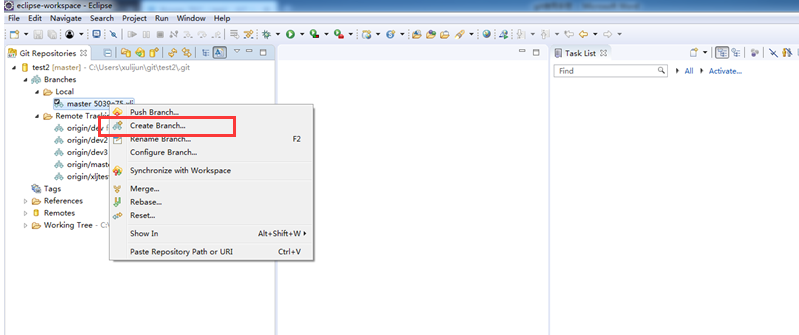


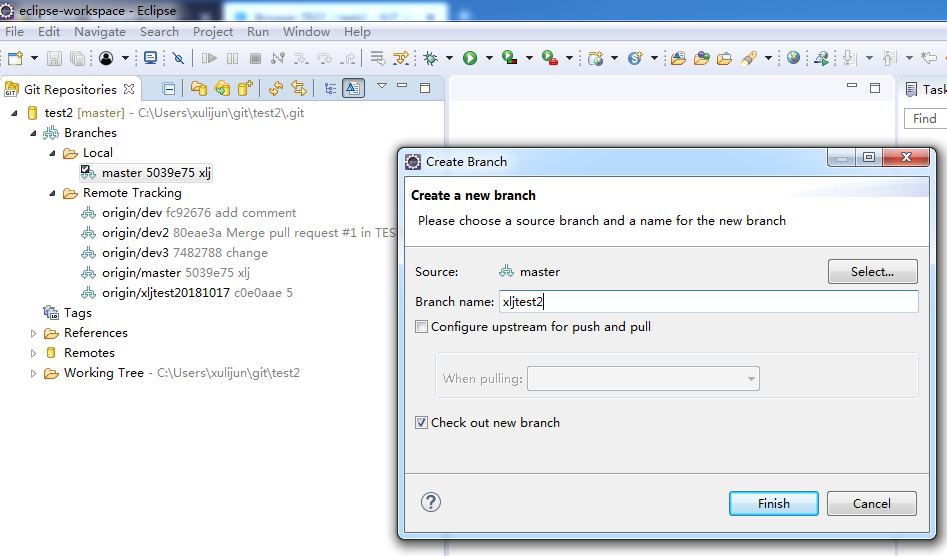
然后就出现创建分支界面，按要求创建即可。



### Eclipse方式：

对应分支右键点击create branch，输入分支名字，创建后push即可。

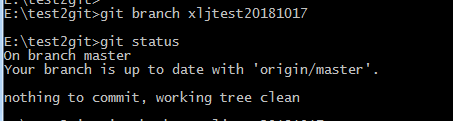




### 命令方式：

git branch branch-name: 创建分支,首先需要到git目录下。注意这是在本地创建了一个分支，并没有push到服务器上。

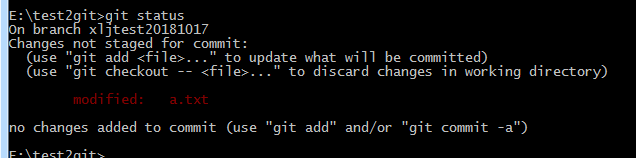
例如： git checkout xljtest20181017



## git status: 查看文件状态

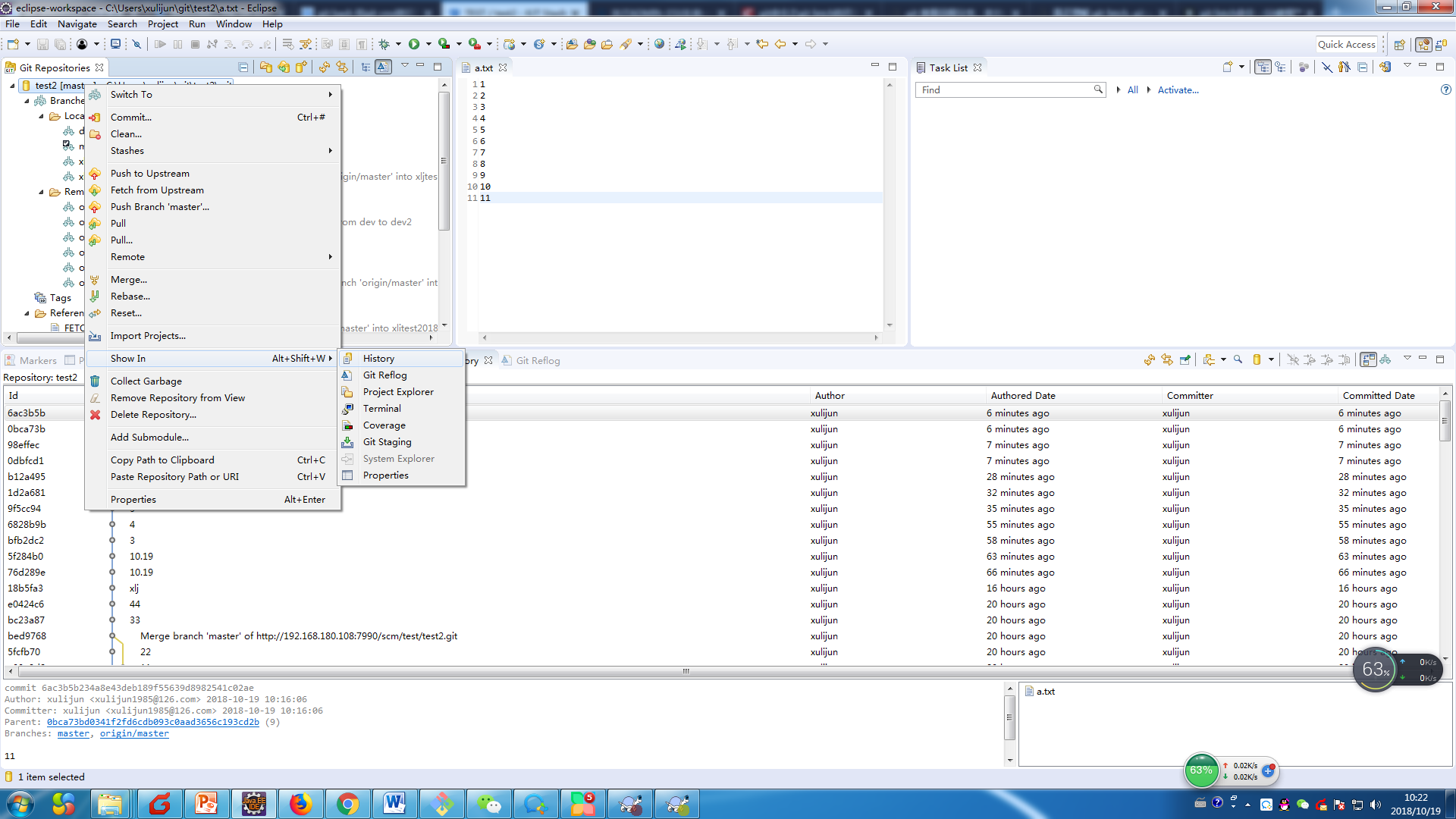
### 命令方式：

本地修改代码后运行git status ，可以显示更改的文件以及一些帮助信息



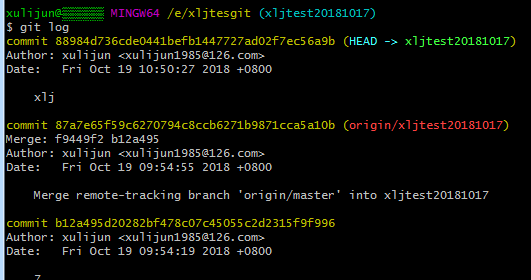
## Git log：显示历史信息

### eclipse方式



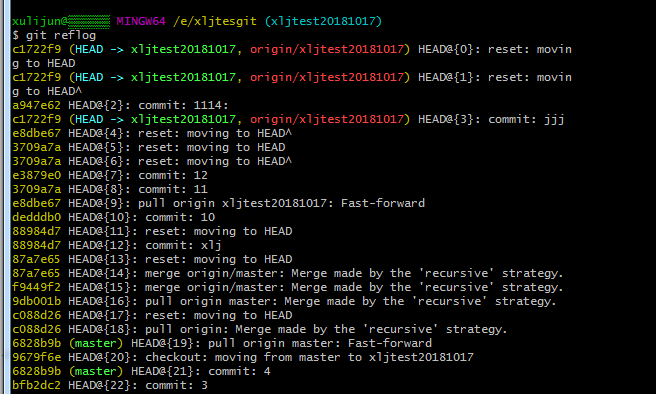
### 命令方式：

Git log



Git reflog

如果在回退以后又想再次回到之前的版本，git reflog 可以查看所有分支的所有操作记录（包括commit和reset的操作），包括已经被删除的commit记录，git log则不能察看已经删除了的commit记录。



## 分支查看

### 命令方式：git branch

git branch –a：显示所有分支

git branch –r：显示远程分支

