**Git应用介绍**

**2017年1月**

目 录

1. Git 功能概述 3

2. Git for Windows软件安装 4

3. Git与远程仓库的关联 5

3.1. 本地Git与oschina的连接 5

3.2. 本地Git与oschina关联 9

3.3. 本地Git功能简介 11

3.4. 本地Git与和团队项目保持同步 12

4. EGit 功能概述 20

4.1. 安装与配置 20

4.2. 创建本地Git仓库 24

4.3. 本地操作 27

4.4. Merge合并冲突 29

4.5. 远程库关联 31

4.6. 远程库同步 34

4.7. 克隆远程库 39

# Git 功能概述

我们用Git来对这些不同的版本进行控制。还可以很方便地查看两个不同版本之间的不同之处。

使用Git，我们只保存最新的一份文件就可以了。

使用Git，我们可以把文件回退到想要的旧版本。

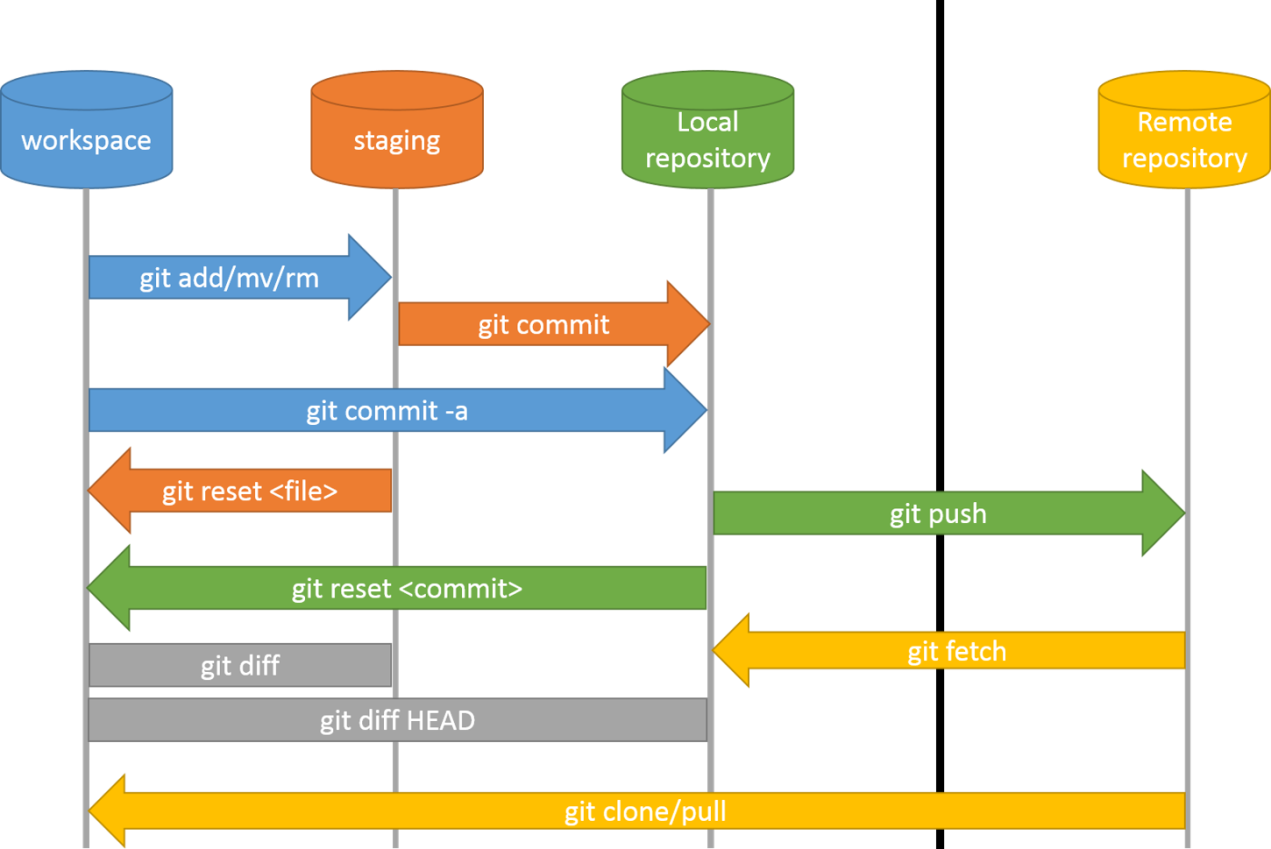
使用Git，我们可以将回退的版本再次还原版本。

Git 有三种状态，你的文件可能处于其中之一：

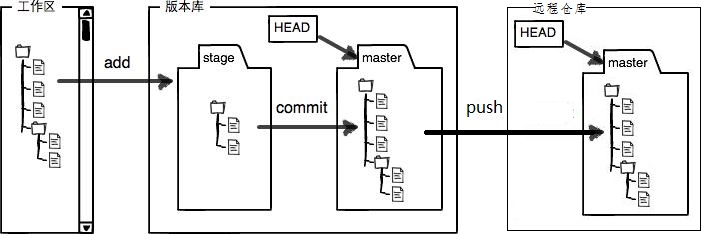
已提交（committed）、已修改（modified）和已暂存（staged）。

* 已提交表示数据已经安全的保存在本地数据库中。
* 已修改表示修改了文件，但还没保存到数据库中。
* 已暂存表示对一个已修改文件的当前版本做了标记，使之包含在下次提交的快照中。

Git本地使用与远程仓库的关系：



图片引用自：[Git introduction for CVS/SVN/TFS users](http://blog.podrezo.com/git-introduction-for-cvssvntfs-users/)



图片引用自：[工作区和暂存区 - 廖雪峰的官方网站](http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000/0013745374151782eb658c5a5ca454eaa451661275886c6000) （做了点修改）

对照查看两张图：

* workspace 即工作区，逻辑上是本地计算机，还没添加到repository的状态；
* staging 即版本库中的stage，是暂存区。修改已经添加进repository，但还没有作为commit提交，类似于缓存；
* Local repository 即版本库中master那个地方。到这一步才算是成功生成一个新版本；
* Remote repository 则是远程仓库。用来将本地仓库上传到网络，可以用于备份、共享、合作。

# Git for Windows软件安装

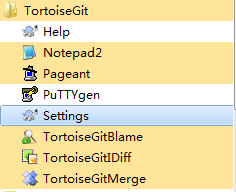
* 在官方网站https://git-for-windows.github.io/workspace 下载最新版本；
* 安装时全部使用默认安装，默认添加环境变量。
* 安装视图化工具TortoiseGit-2.3.0.0-64bit.msi
* 安装视图化工具语言包TortoiseGit-LanguagePack-2.3.0.0-64bit-zh\_CN.msi

# Git与远程仓库的关联

## 本地Git与oschina的连接

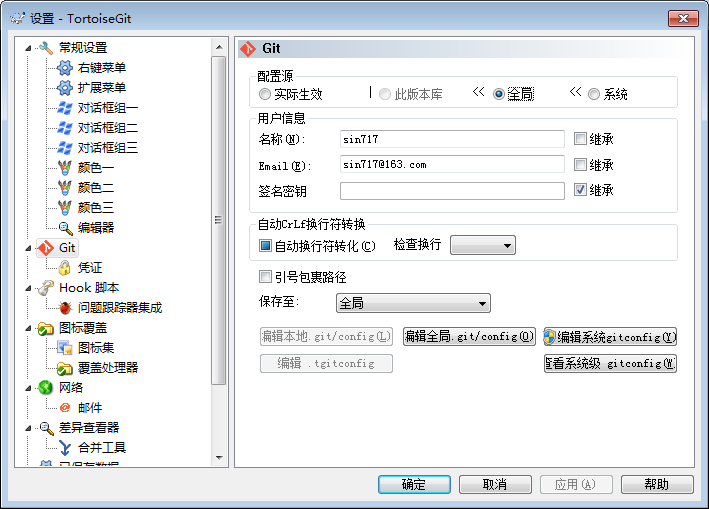
1. 到oschina注册账号。
2. 本地配置

打开：开始 -> 所有程序 -> TortoiseGit -> Settings



配置源：全局

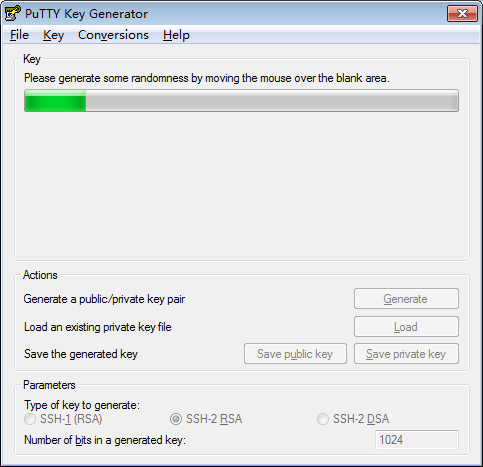
用户信息：步骤1中注册色用户名和邮箱

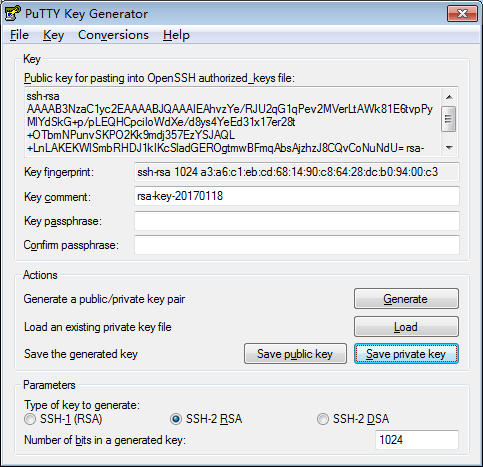


1. 生成ssh key

打开：开始 -> 所有程序 -> TortoiseGit -> Puttygen

设置密钥长度为1024，点击“Generate”，鼠标需要在进度条上上下滑动。



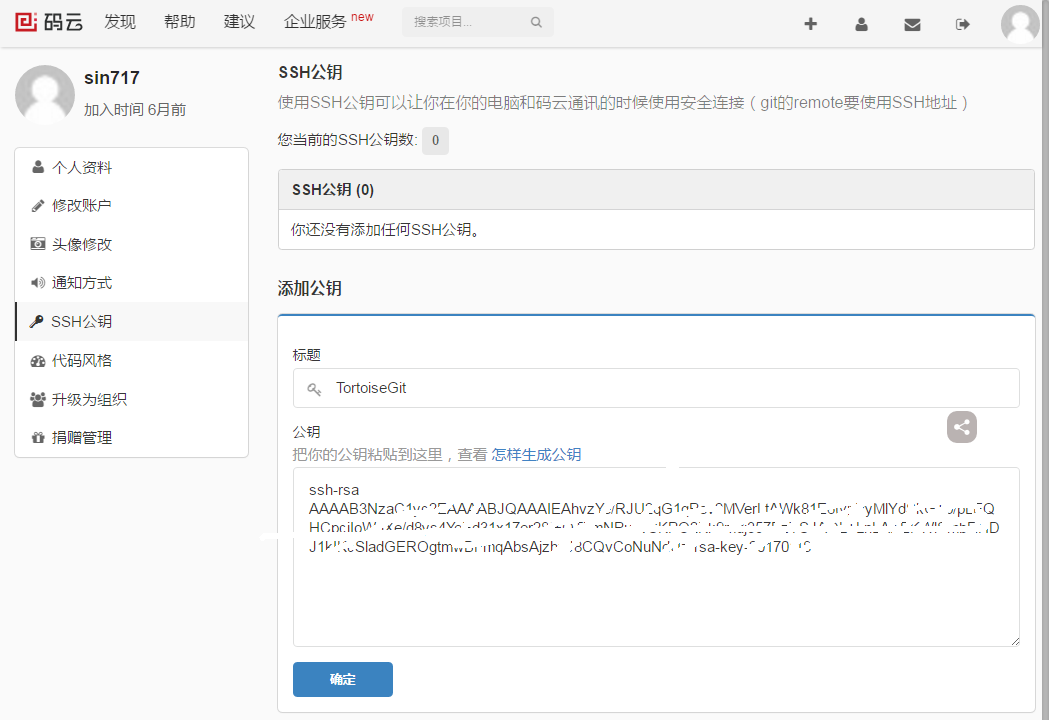


保存公钥和密钥；

将生成的ssh key复制到剪贴板，保存的公钥文件中获取。

1. 打开oschina，进入个人资料修改“SSH公钥”：

将ssh key粘贴到右边的公钥里面。标题随便命名即可。



点击下面的确认，输入验证密码就添加成功了。

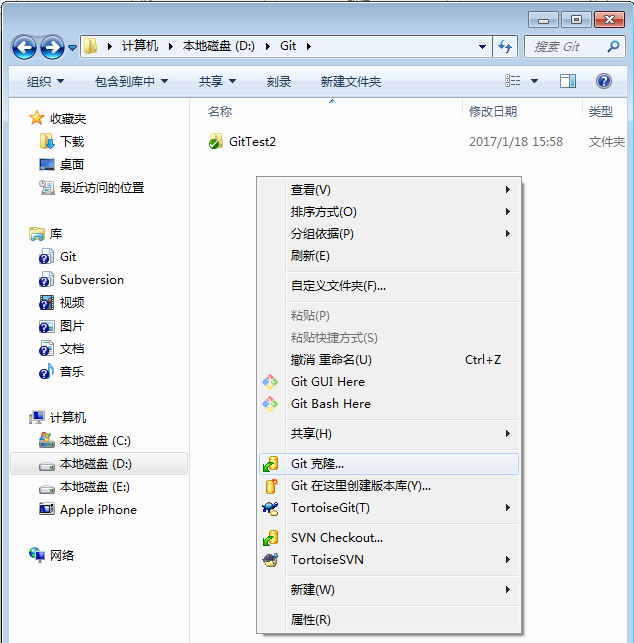
## 本地Git与oschina关联

1. 获取远程库SSH地址

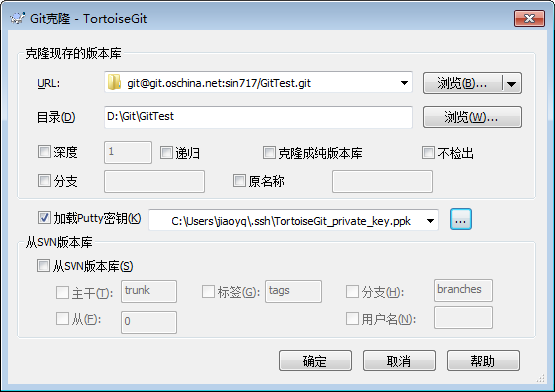


1. 克隆远程库

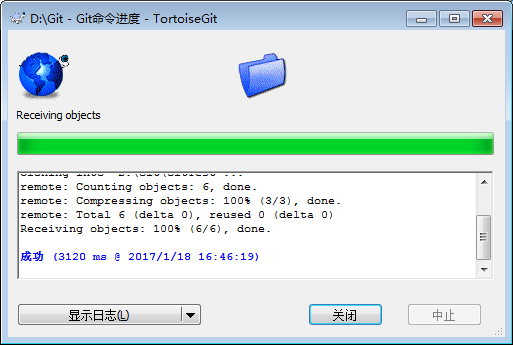
在需要创建Git本地仓库的目录下，右键 -> Git 克隆…。



把SSH地址粘贴到URL，导入之前创建的密钥文件，确认。



克隆完成关闭。



## 本地Git功能简介

* 添加(add)：



将从未监控的文件提交暂存区

* 还原：



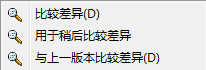
将文件从暂存区恢复到工作区或将修改的文件还原到最后提交版本

* 提交(commit)：



将暂存区的内容提交到本地库

* 比较(diff)：

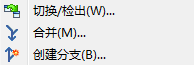


比较差异：将当前文档与最后提交版本内容比较，当前文件夹内容与最后提交版本比较

用于稍后比较差异：将当前文件作为比较基础，与随后选择的文件进行比较

与上一版本比较差异：将当前文档或当前文件夹内容的最后2次提交的内容进行比较

* 分支(branch)：



创建分支：创建新分支，并且切换过去

合并：将制定分支内容合并到当前分支内，被制定的分支不会被删除

切换/检出：切换分支

* 日志(log)：



查看个提交版本内容，以及切换版本内容。

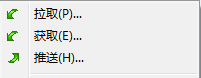
## 本地Git与和团队项目保持同步

本地Git和远程仓库的同步有3个命令：拉取(pull)、获取(fetch)、推送(push)

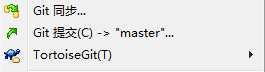
拉取(pull)：相当于是从远程获取最新版本并merge到本地；

获取(fetch)：相当于是从远程获取最新版本到本地，不会自动merge；

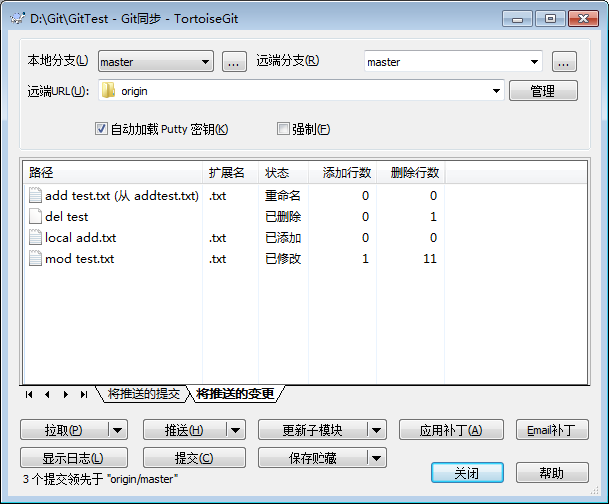
推送(push)：将本地更新推送到远程。



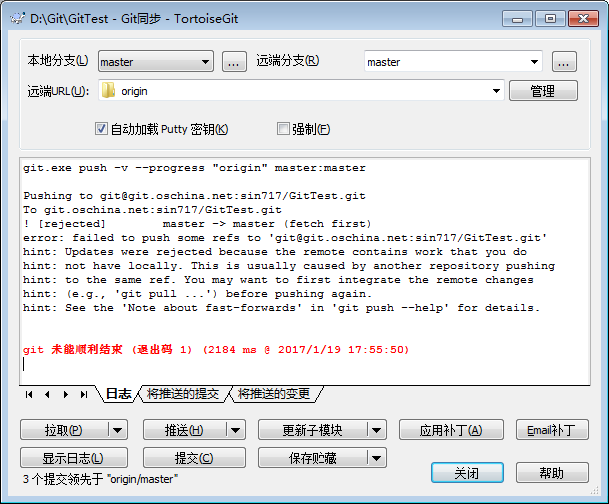
1. 右键 -> Git 同步…



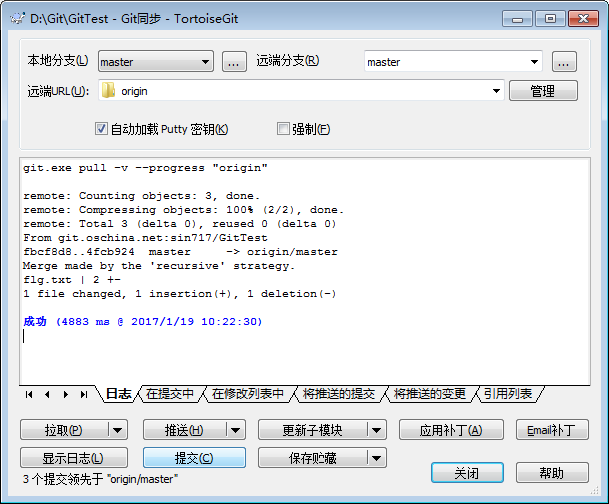
1. 画面显示本次同步操作，本地文件变更情况。点击“推送”



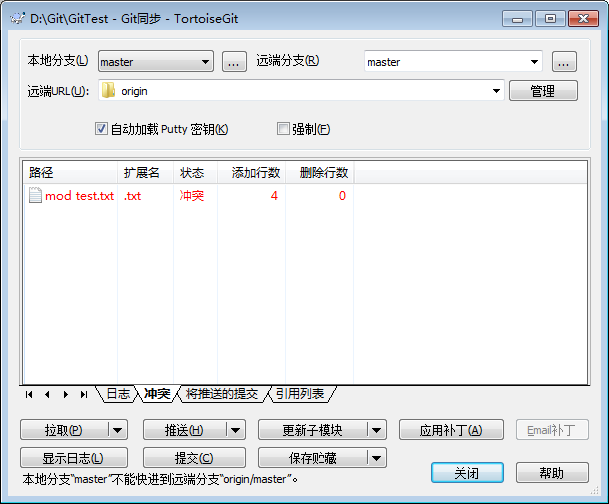
1. 因远程库发生变更未更新本地库，造成推送失败，重新拉去远程库更新，点击“拉取”



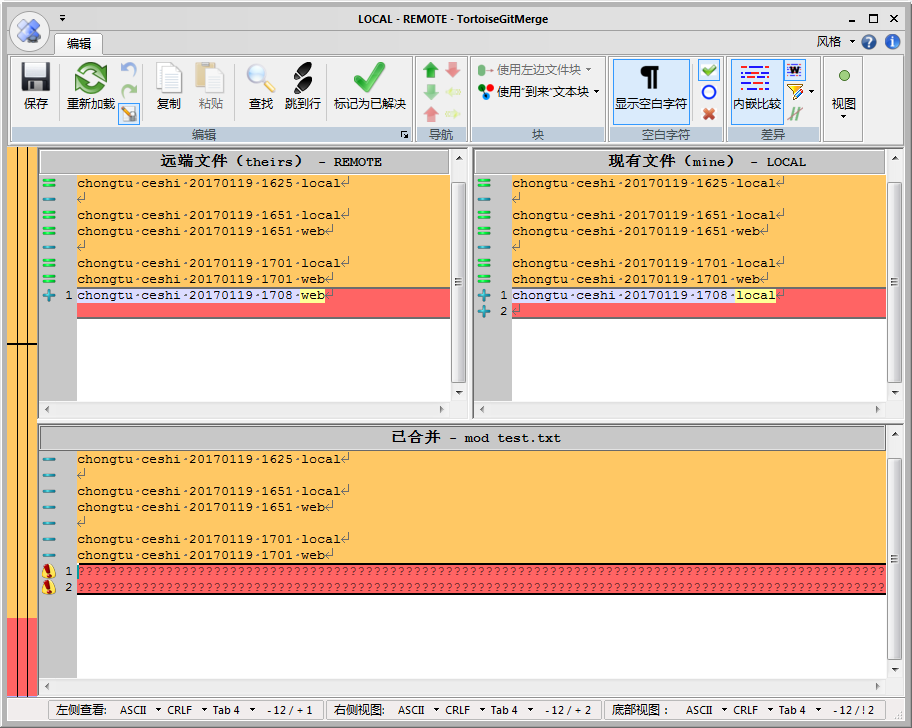
1. 拉取成功，重新推送本地文件

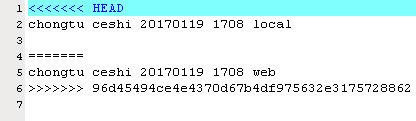


1. 拉取成功，但发生文件冲突，如下图。



可以使用 TortoiseGit自带的比对软件，也可以手动修改文件内容：



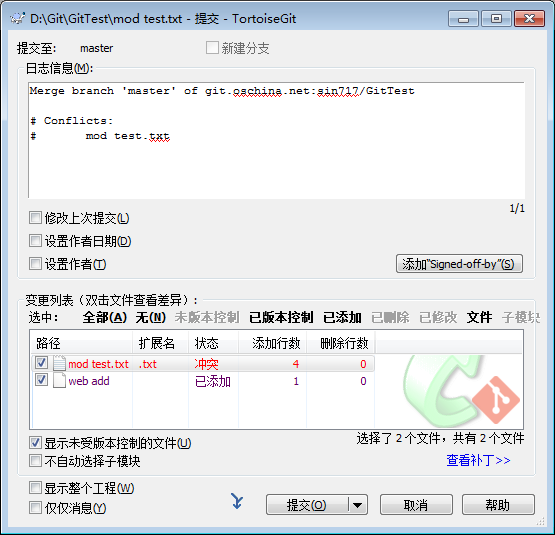


红框中的“=======”是冲突的分割线。

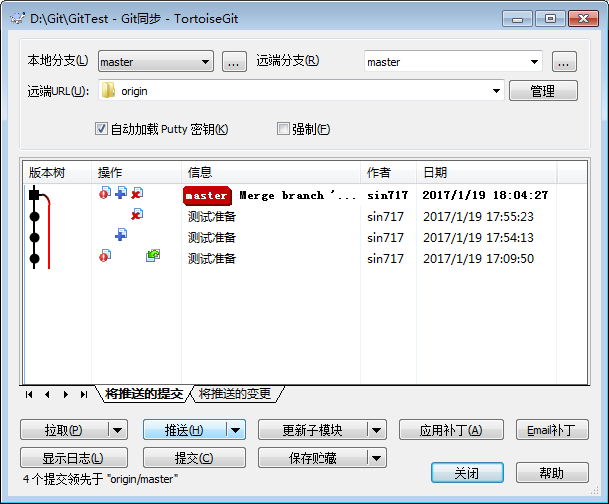
“<<<<<<< HEAD”和分割线之间的是本地的文本，分割线和“>>>>>>> xxxxx”之间的是远程分支的文本。

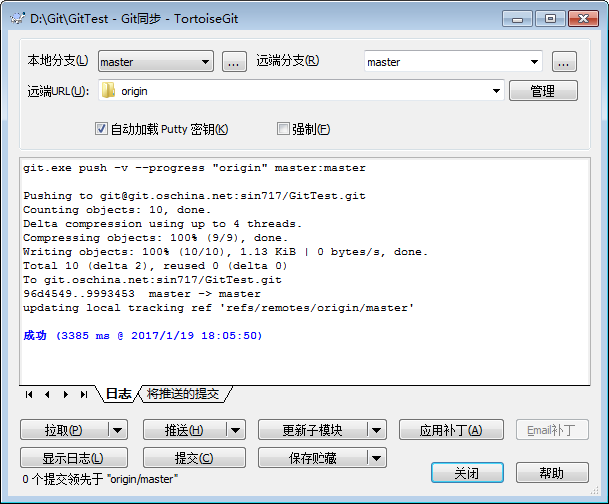
冲突内容解决后，只需将冲突符号删除。

1. 冲突解决后点击提交，将远程库的内容和本地库的内容完成merge。



1. 再次执行推送，完成本地Git与远程库的同步

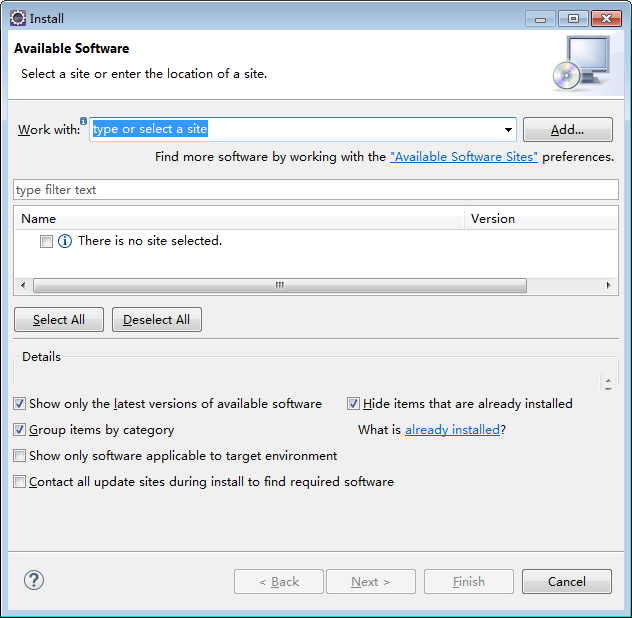




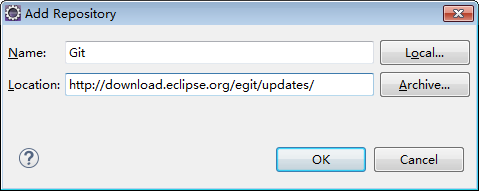
# EGit 功能概述

## 安装与配置

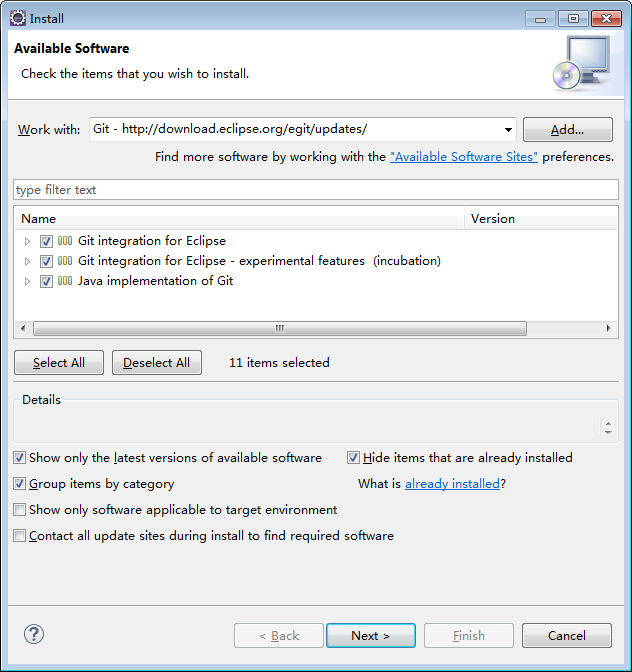
* 1. 在Eclipse菜单中打开help -> Install new software，在弹框中点击“Add”；



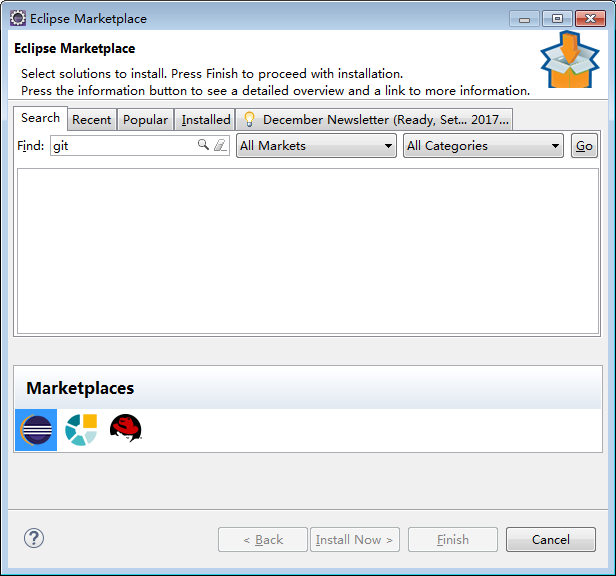
在弹框中输入Git下载地址：http://download.eclipse.org/egit/updates/，点击“OK”；



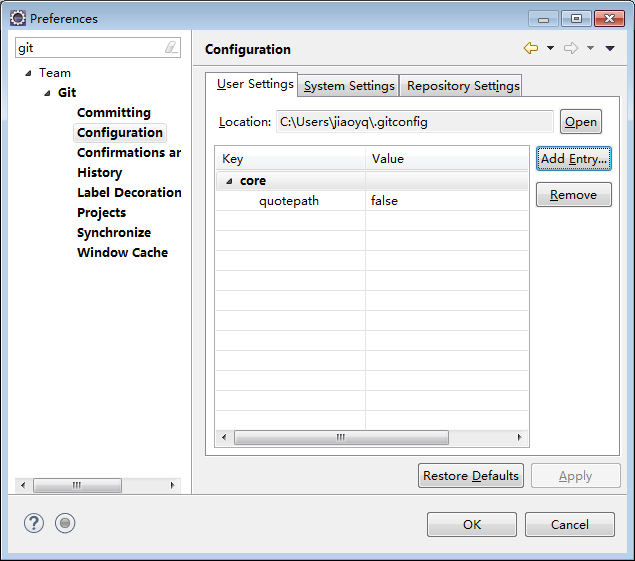
少许等待后，画面会显示Git的安装程序，全选后，点击“Next”执行安装。



* 1. 也可以在Eclipse菜单中打开help -> Eclipse Marketplace，在弹框中搜索EGit；



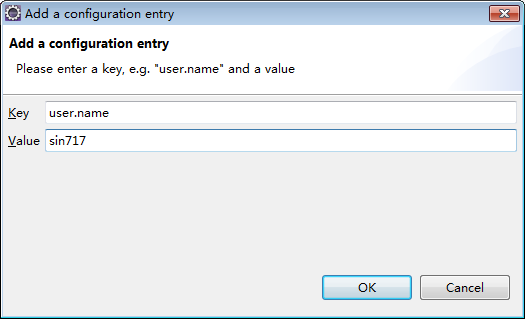
* 1. 安装完成后在Window -> Preferences -> Team -> Git -> Configuration -> User Settings进行账户配置。点击“Add Entry”



* 1. 在弹框中输入用户名和邮箱

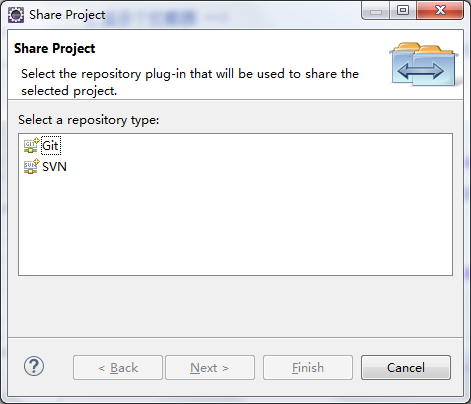
Key：user.name Value：用户名

Key：user.email Value：邮箱地址

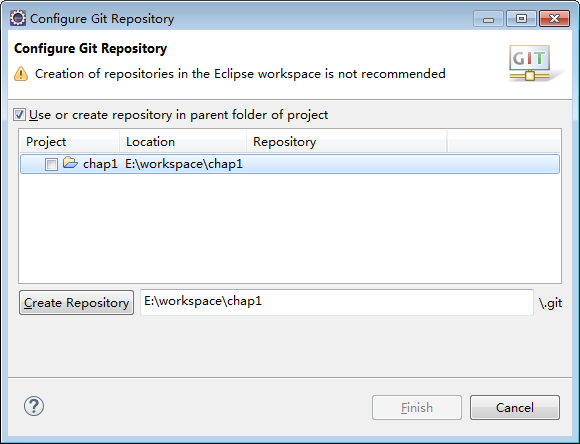


## 创建本地Git仓库

* 1. 在Eclipse中创建一个项目
  2. 项目名右键选择Team > Share Project 选择GIT



* 1. 创建本地仓库

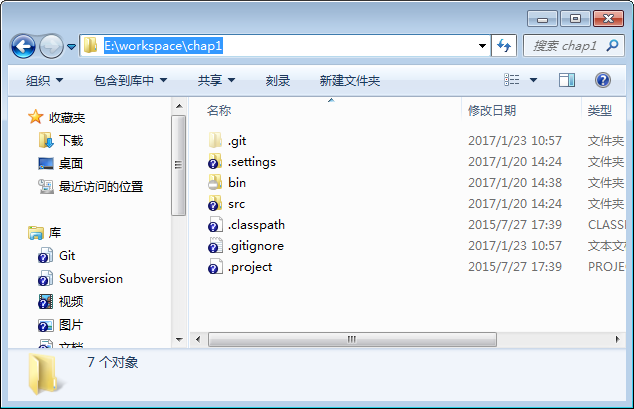


3.输入仓库地址后点击创建本地仓库

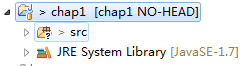
2.选中

1.勾选

* 1. 创建成功后再项目文件夹内生成一个隐藏文件夹：.git；

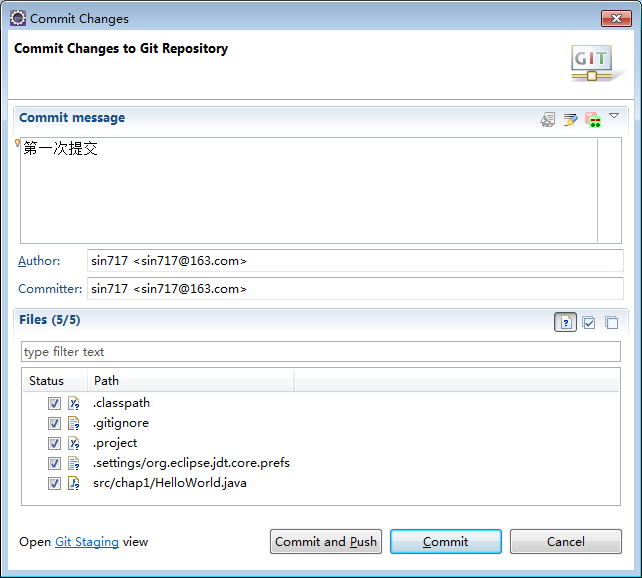


* 1. Eclipse中的项目也建立Git版本控制，但没有创建分支。

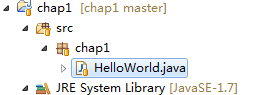


文件夹中的符号”?”表示此文件夹处于untracked状态，这样就成功创建GIT仓库。

* 1. 提交项目内容：Team > Commit



* 1. 首次提交后，会自动生成master分支。在红框的位置可以看到当前的分支。



## 本地操作

* 添加(add)：



将从未监控的文件提交暂存区

* 还原：



将文件从暂存区恢复到工作区

* 提交(commit)：



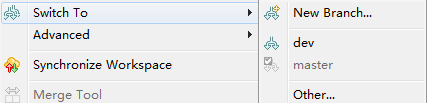
将暂存区的内容提交到本地库

* 提交(commit)：



将暂存区的内容提交到本地库

* 分支(branch)：



New Branch：创建新分支，并且切换过去

分支名：切换分支，切换成功后项目名称后面显示当前分支



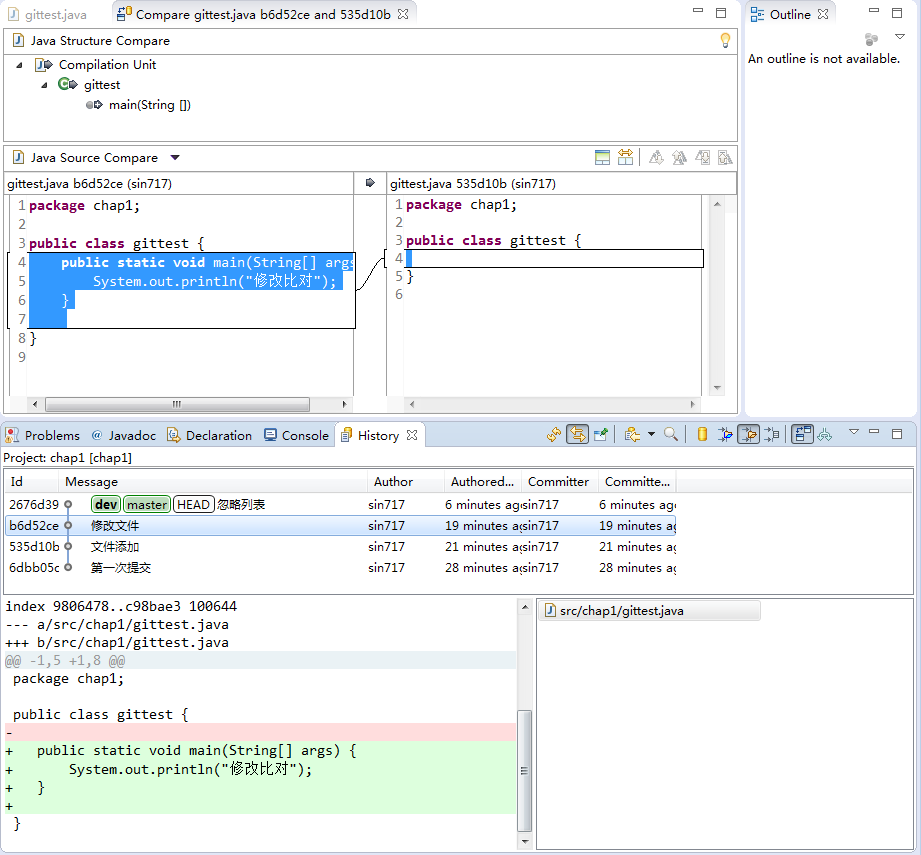
* 仓库管理：



* 日志(log)：



查看个提交版本内容，文件变更比对以及切换版本内容。



选择比对模式后，双击比对的文件，显示比对内容

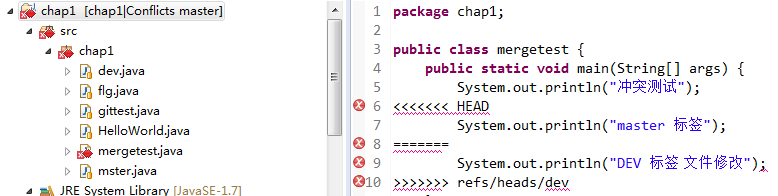
* 合并(Merge)：



制定分支与当前分支合并，修改当前分支，保留指定分支。

## Merge合并冲突

* 冲突提示如下图：



项目名后的CONFLICT表示有冲突，冲突文件图标有红色×和双向箭头标注。

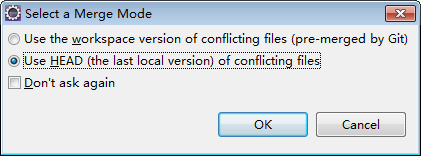
红框中的“=======”是冲突的分割线。

“<<<<<<< HEAD”和分割线之间的是本地的文本，分割线和“>>>>>>> refs/heads/dev”之间的是远程分支的文本。

* 解决方法

右键冲突文件选择Team > Merge Tool，修改source后保存。

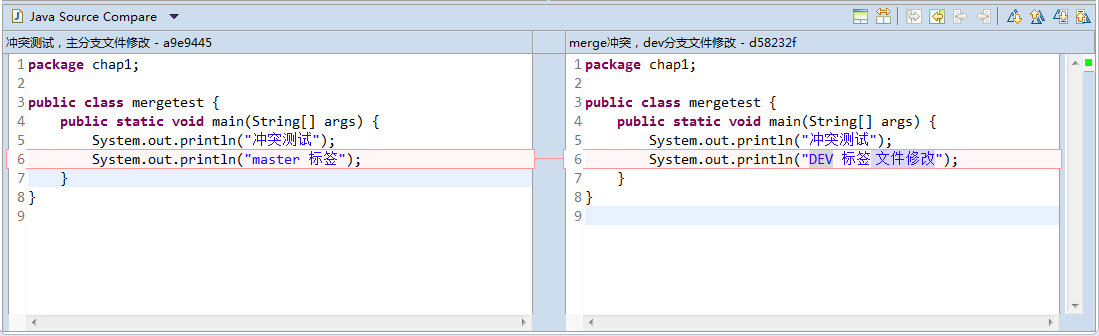




第一项是将GIT自动合并过的本地文件和对方文件进行对比

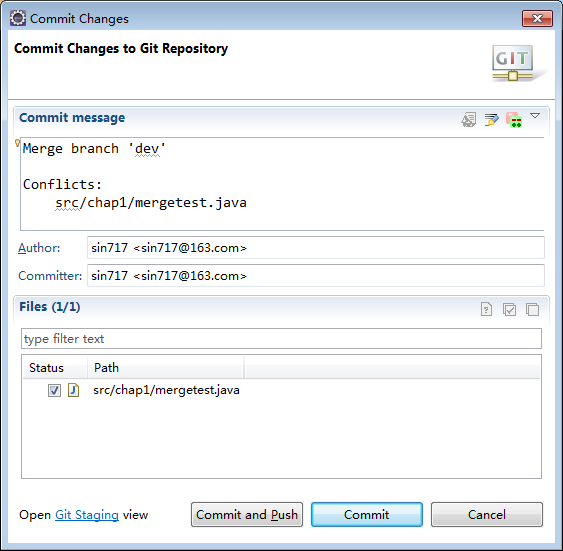
第二项是用本地最新版本的文件和对方文件进行对比，建议用此项

比对界面：



也可以在原source中修改。

修改完成后添加(add)、提交(commit)即可：

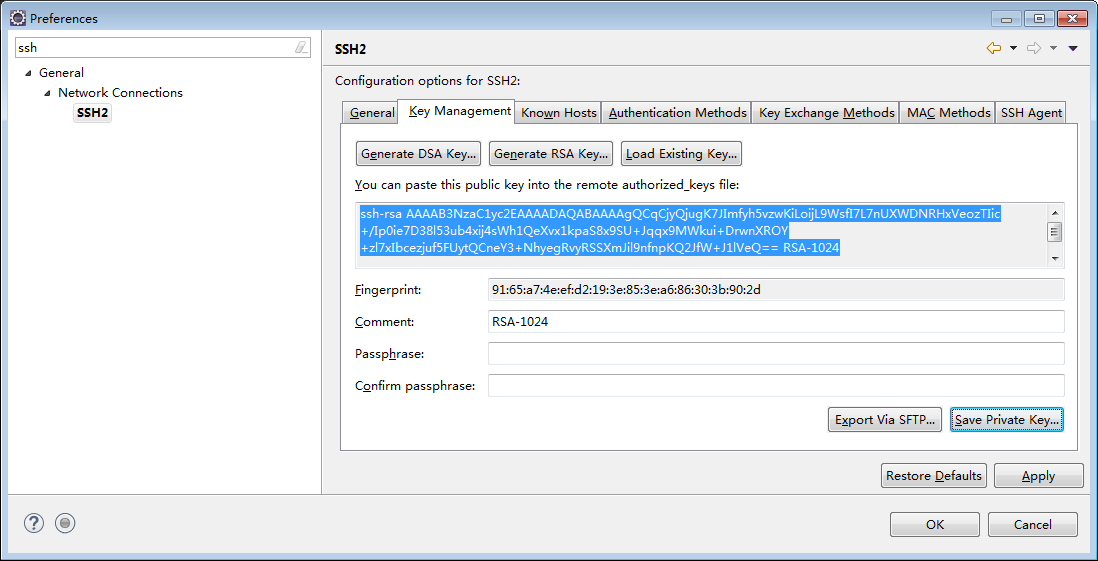


## 远程库关联

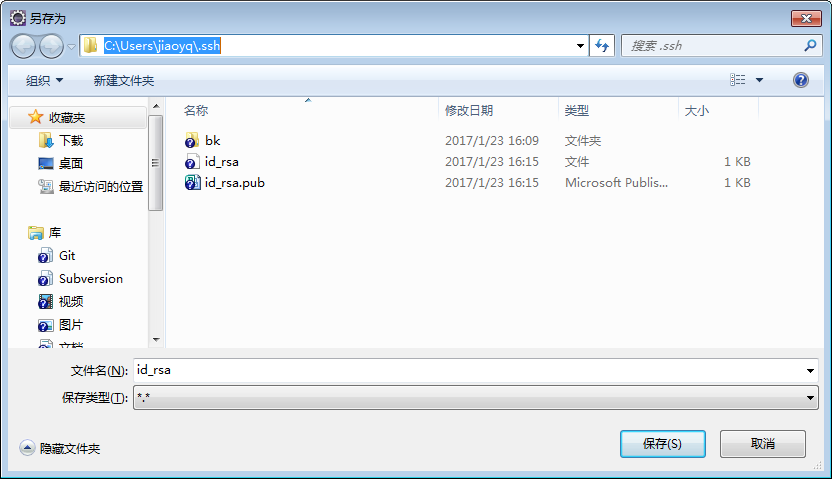
* 1. 生成SSH Key：

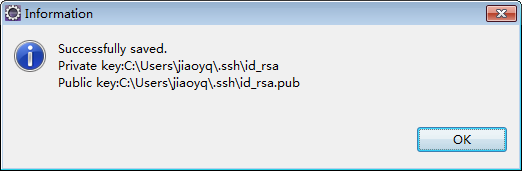
打开Window -> Preferences -> General -> Network Connections -> SSH2；

选择Key Management，点击“Generate RSA Key…”，将生成的ssh key复制到剪贴板。

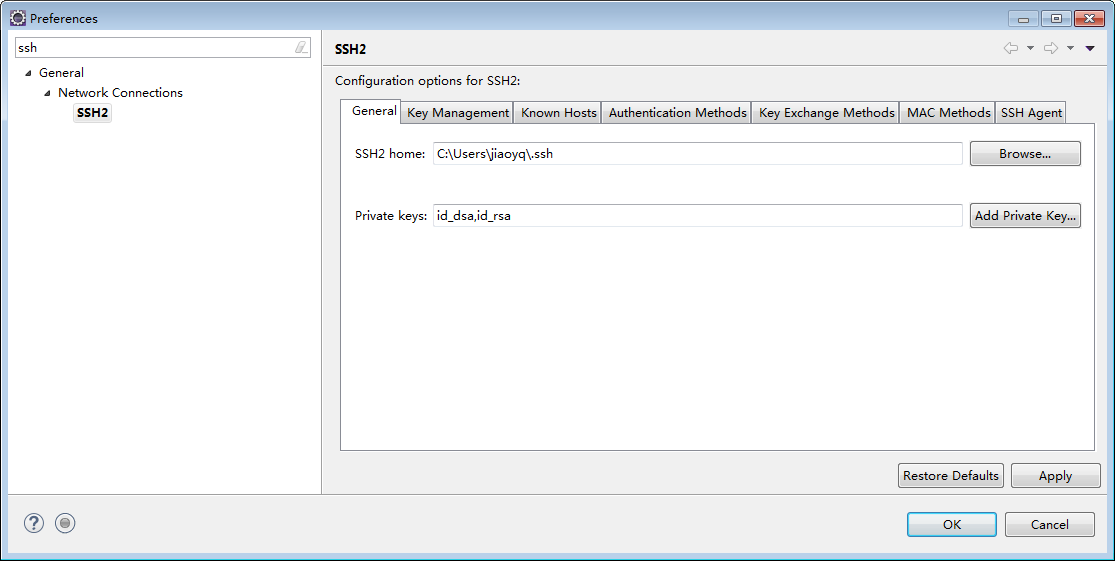


点击“Save Private Key…”保存SSH Key；



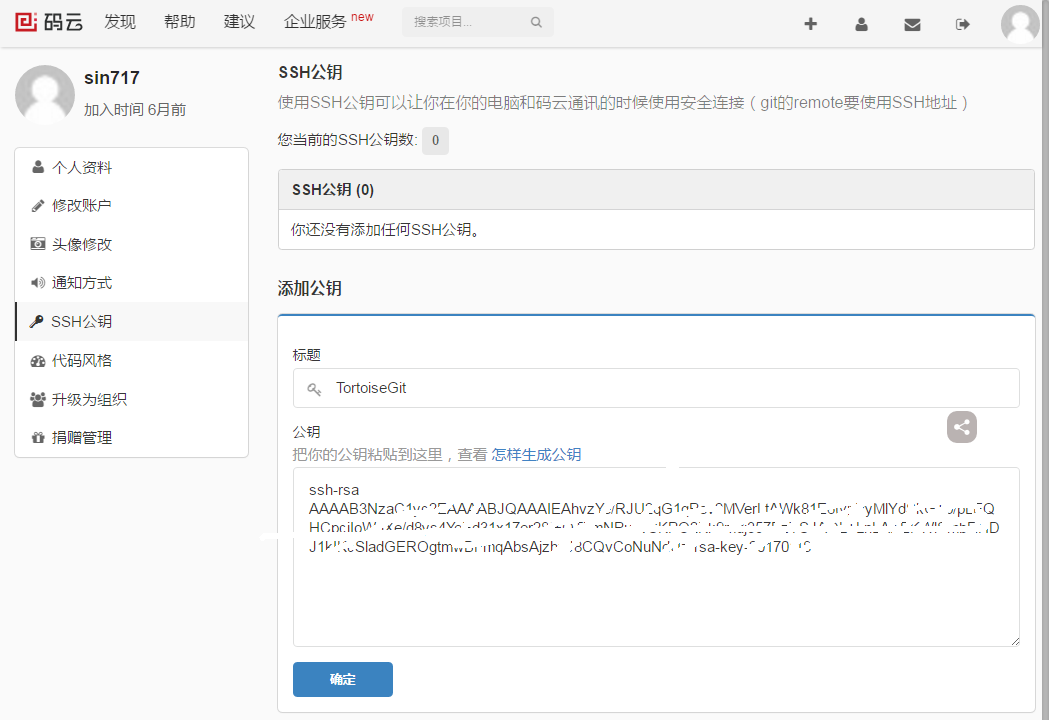


选择General，选择之前保存的SSH Key文件后点击“OK”。



* 1. 打开oschina，进入个人资料修改“SSH公钥”：

将ssh key粘贴到右边的公钥里面。标题随便命名即可。



* 1. 点击下面的确认，输入验证密码就添加成功了。

## 远程库同步

* 1. 获取远程仓库SSH地址：



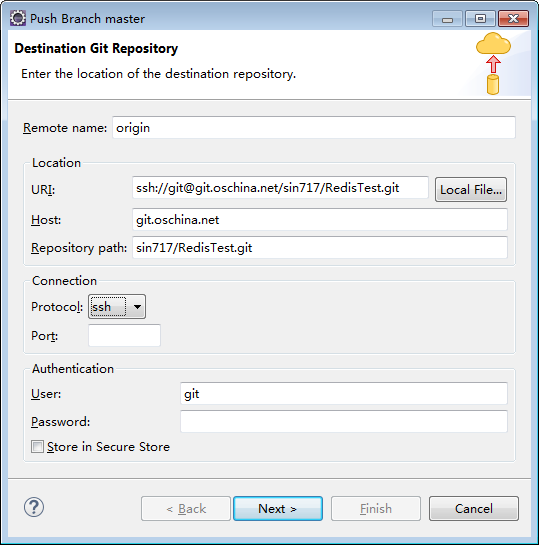
* 1. 本地库文件上传远程库（失败）

选择项目右键选择：Team -> Push Branch ‘master’

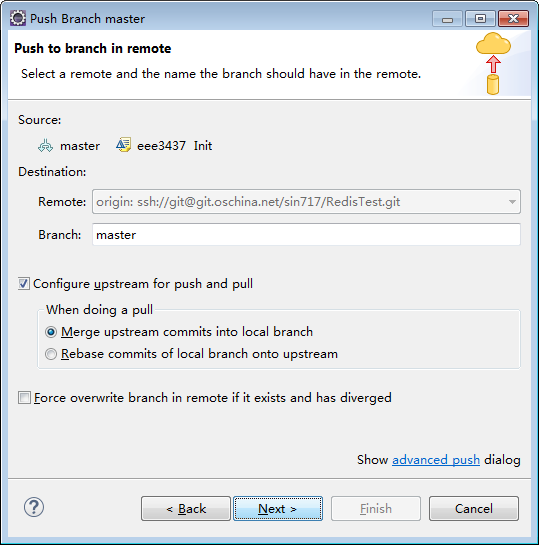


设置远程库上传信息：

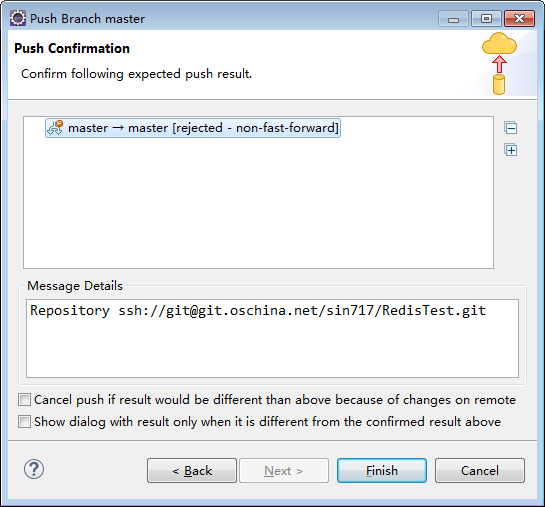
* + 1. URI：粘贴远程库地址，其他信息会自动显示无需修改；
    2. Protocol：选择ssh，URI会发生自动变更。
    3. 点击“Next”



确认上传版本和远程库地址，直接点击“Next”



上传请求被拒绝，本地库无远程库版本信息。点击“Finish”本次操作结束。

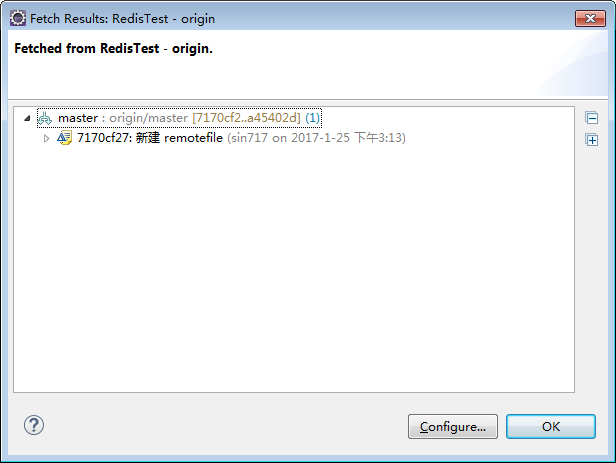


* 1. 获取全程库信息

选择项目右键选择：Team -> Fetch from Upstream

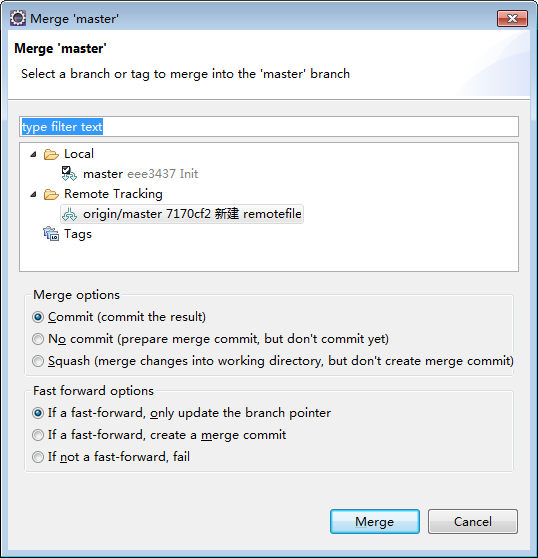


弹出框显示远程库信息，点击“OK”



选择项目右键选择：Team -> Merge，

在弹出框中我们看到之前我们从远程库获取的最新版本。点击“Merge”。



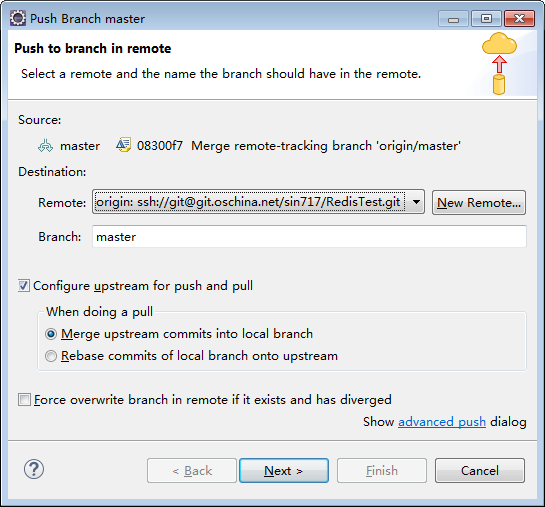
* 1. 本地库文件上传远程库（成功）

选择项目右键选择：Team -> Push Branch ‘master’

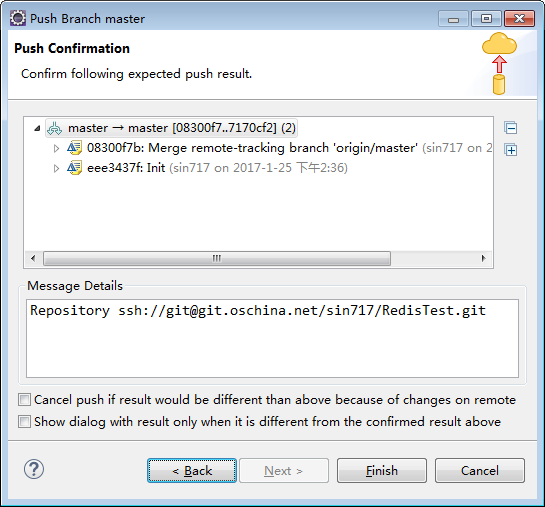


因之前已在这个项目中设置了远程库的信息，二次上传不做设置，

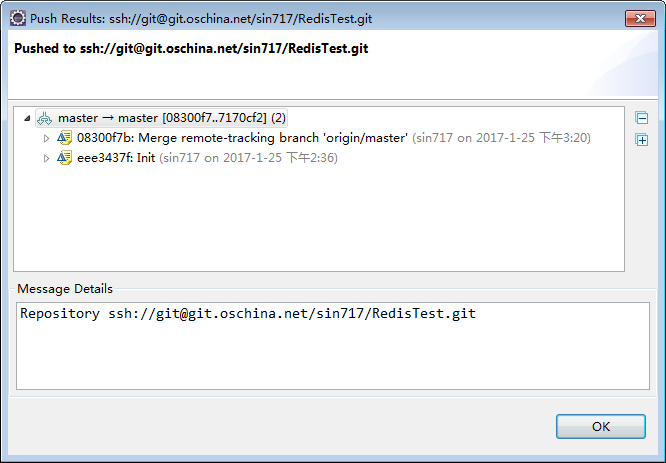
画面直接显示确认上传最新版本和远程库地址，直接点击“Next”



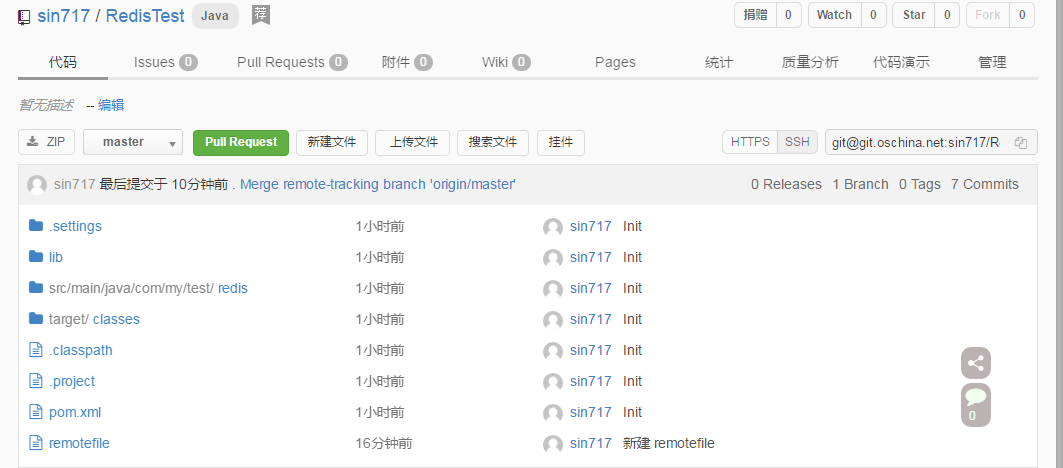
上传内容确认，点击“Finish”开始上传。



上传成功



* 1. 远程库确认



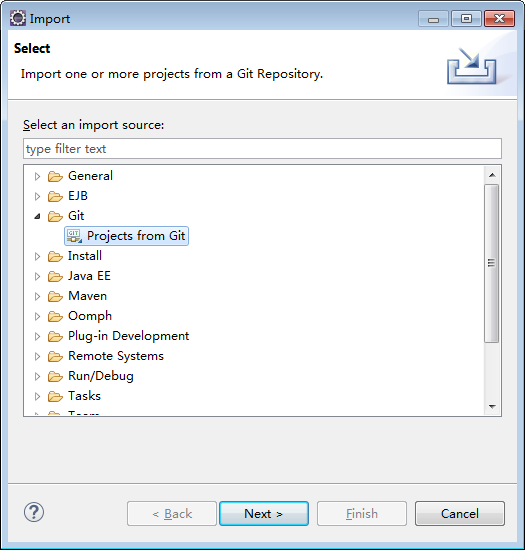
## 克隆远程库

* 1. 获取远程仓库SSH地址：

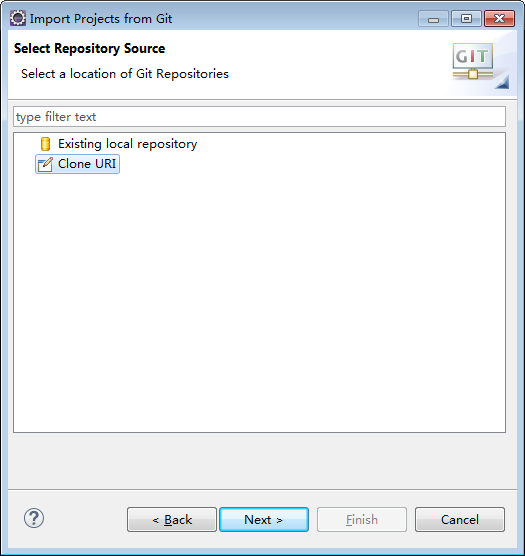


* 1. 克隆远程库

打开菜单：File -> Import，选择git视图中克隆代码库，点击“Next”

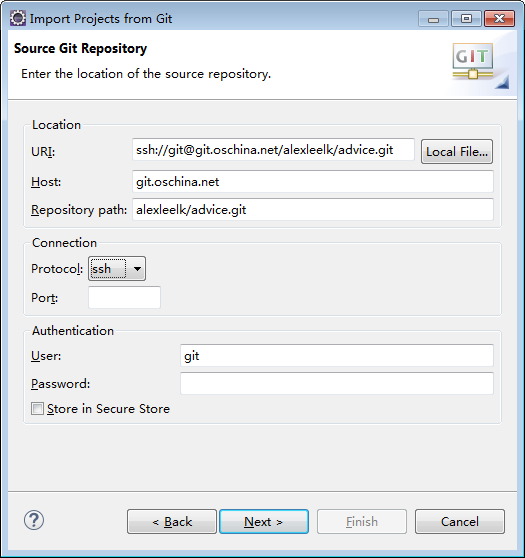


选择“Clone URI” ，点击“Next”

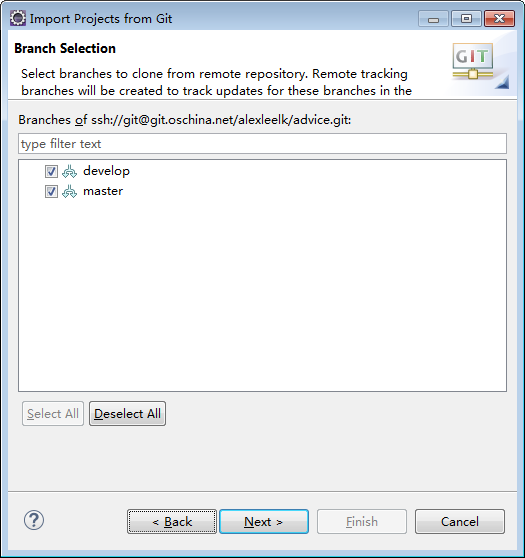


设置远程库信息：

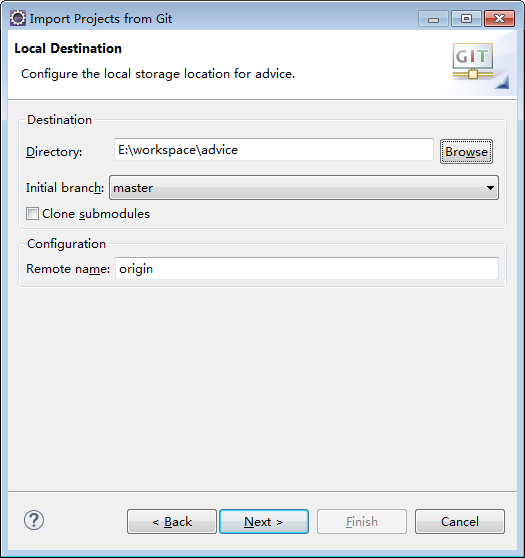
* + 1. URI：粘贴远程库地址，其他信息会自动显示无需修改；
    2. Protocol：选择ssh，URI会发生自动变更。
    3. 点击“Next”



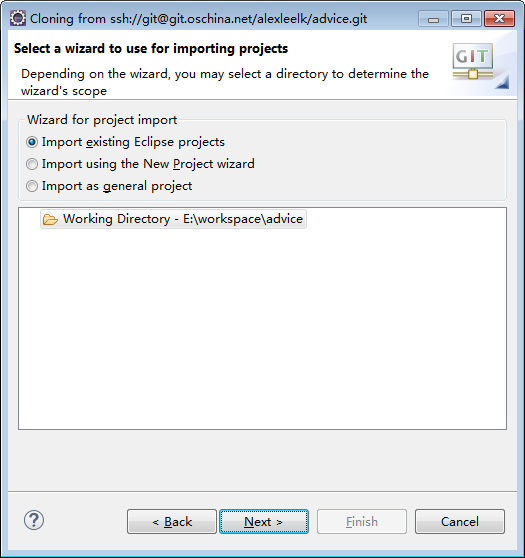
选择远程库下载的分支，默认全选，点击“Next”



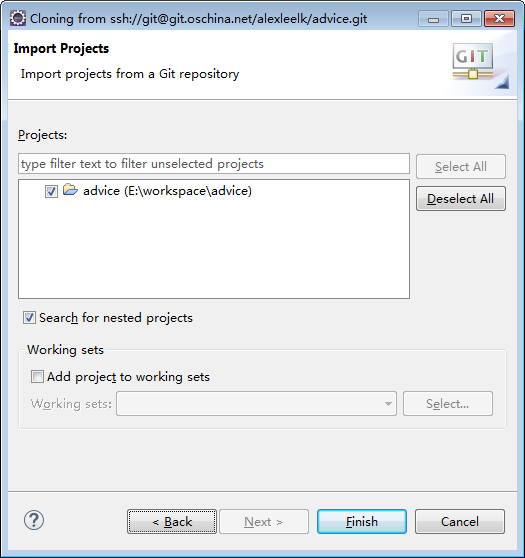
选择本地文件保存位置，点击“Next”



默认选择，点击“Next”



默认选择，点击“Finish”



导入成功

