## git、gerrit的使用方法和规范

### 1、新员工git安装环境准备

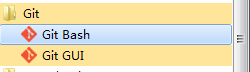
首先从服务器端

<ftp://192.168.31.10/Software/Tool/Git/>

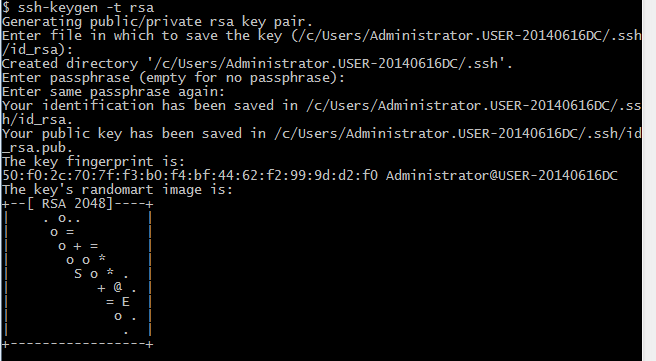
（用户名/密码 paypalm/paypalms）获取软件Git-1.9.4-preview20140929

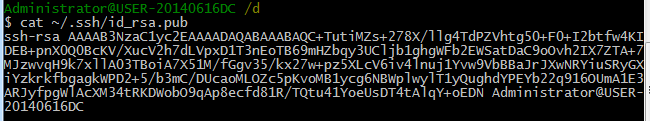
#### 1、默认安装

Git-1.9.4-preview20140929

安装完成后打开git bash编辑器

生成密钥对：ssh-keygen -t rsa 按三次回车键，默认生成路径如下图

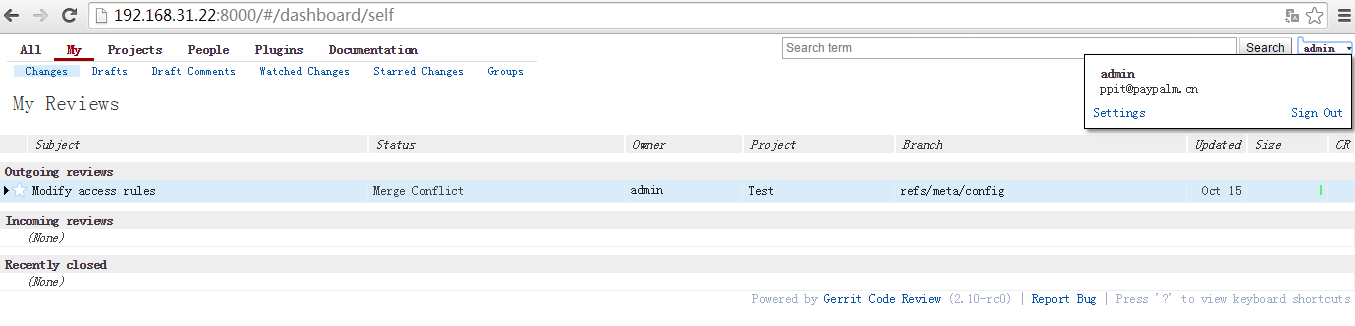


将生成的公钥内容在gerrit中进行添加（参考下文gerrit注册使用） 每个人不同环境可以添加多个对应的公钥

### 2、gerrit注册使用

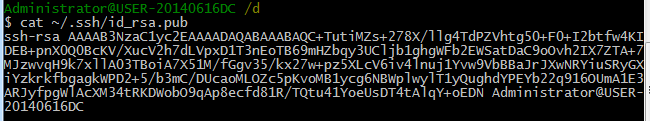
#### 1、申请账号

[通过邮件向PPCM@paypalm.cn](mailto:通过邮件向PPCM@paypalm.cn)发邮件申请，打开gerrit网站（http://192.168.31.10:8088），登录后在右上角进行setting设置

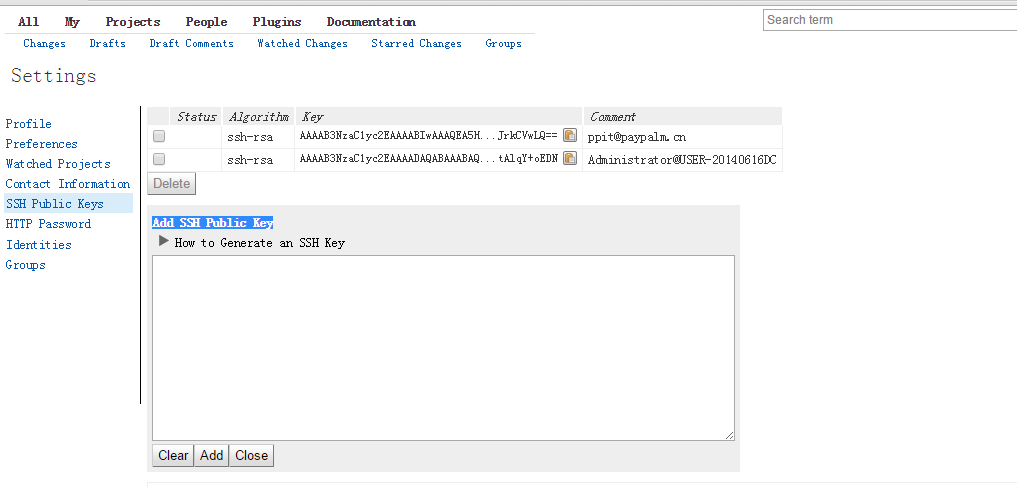


#### 2、公钥添加

点击SSH Public Keys》Add Key选项进行公钥添加







#### 3、邮箱注册

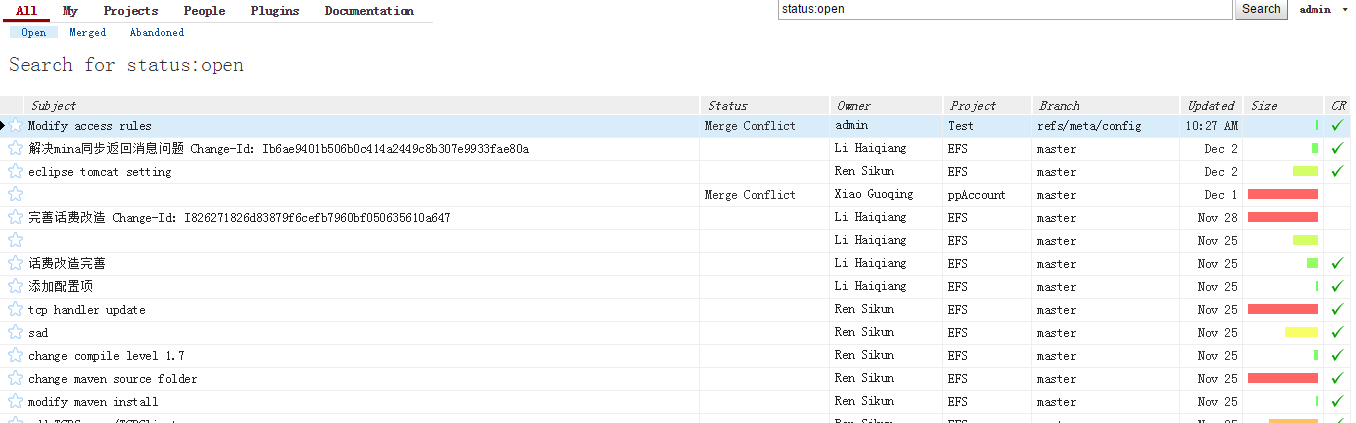
点击Register New Email 进行邮箱注册，注册后有邮件发送至你的邮箱点开链接重新登录



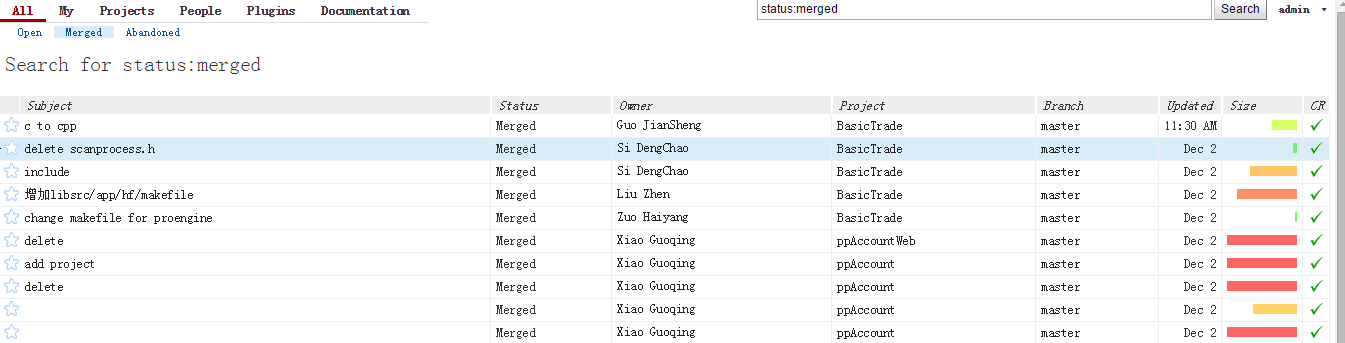
### 3、gerrit主要功能介绍

#### 1、常规功能

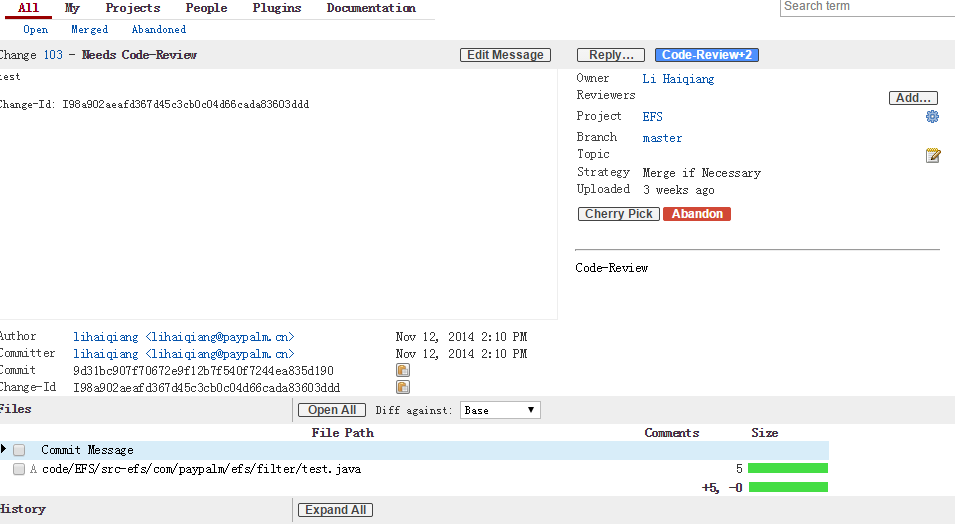
1、登录gerrit》ALL》open状态，此显示为已推送但还没有入库的所有patch，CR状态栏中绿色对勾代表已评审状态，可以根据计划入库

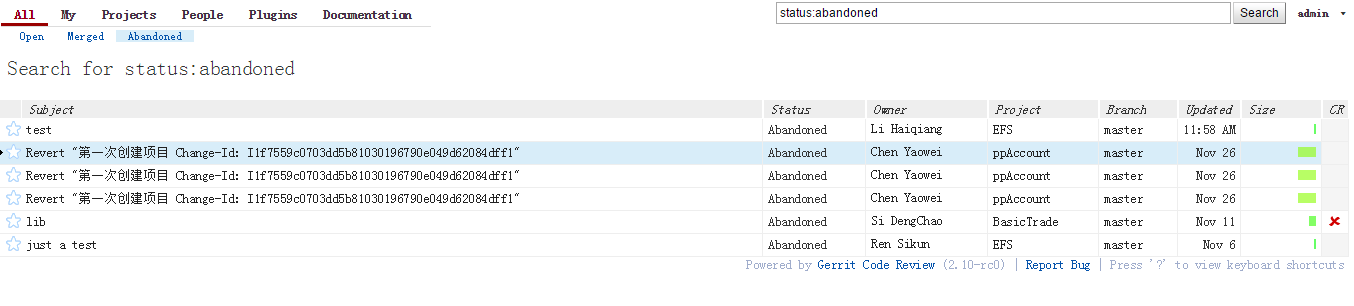


2、gerrit》ALL》Merged状态表示所有已经进入项目库的patch

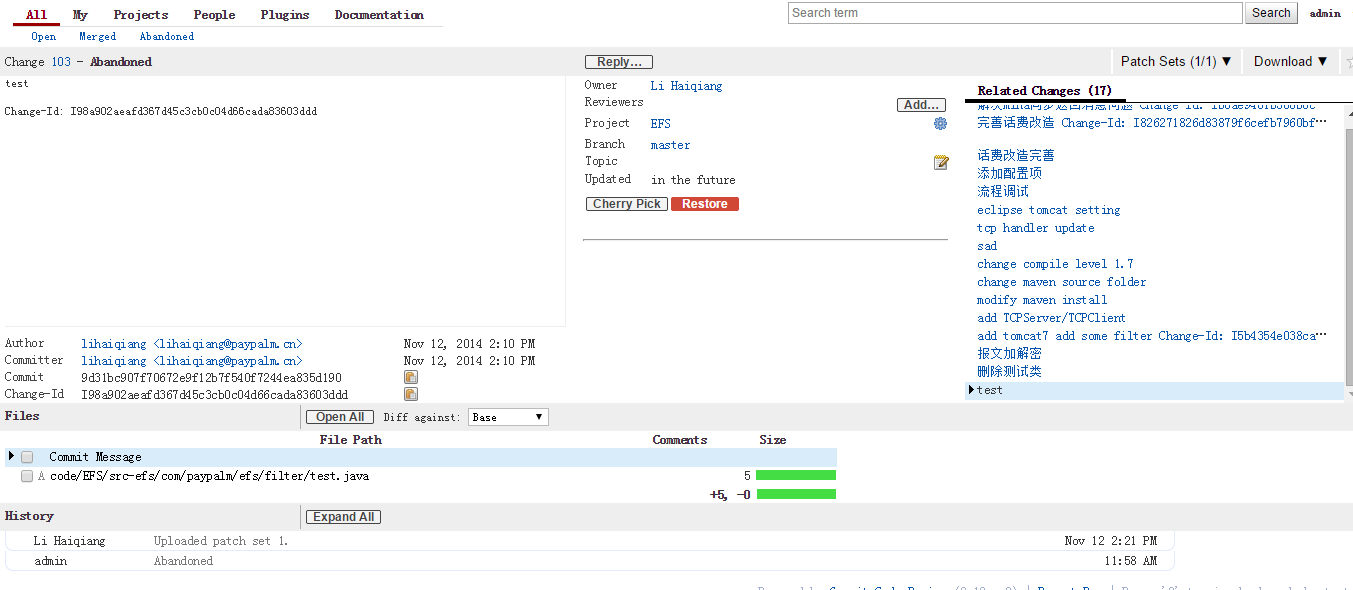


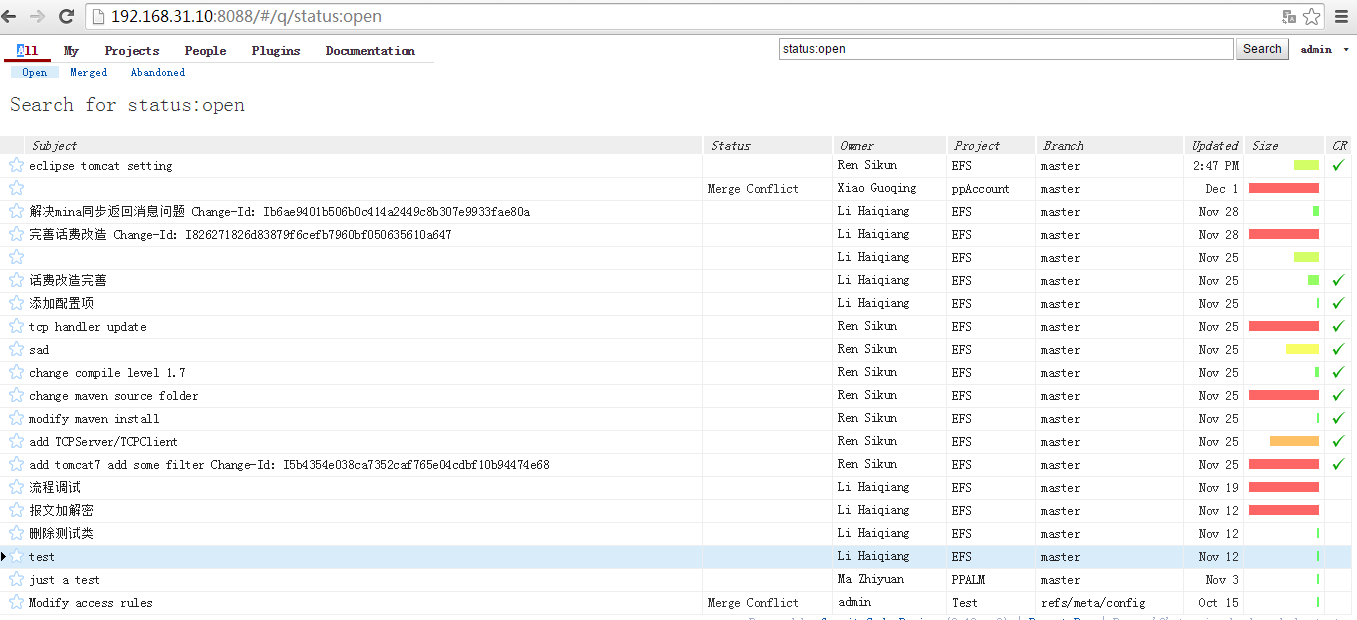
3、提交patch后，开发人员可能觉得不太满意会选择放弃，gerrit》ALL》Abandoned即为已放弃的patch，只有还没有入库的patch才能选择放弃，点击进入patch，橘黄色Abandon即为放弃选项，放弃后的patch依然可以进行还原，如以下操作



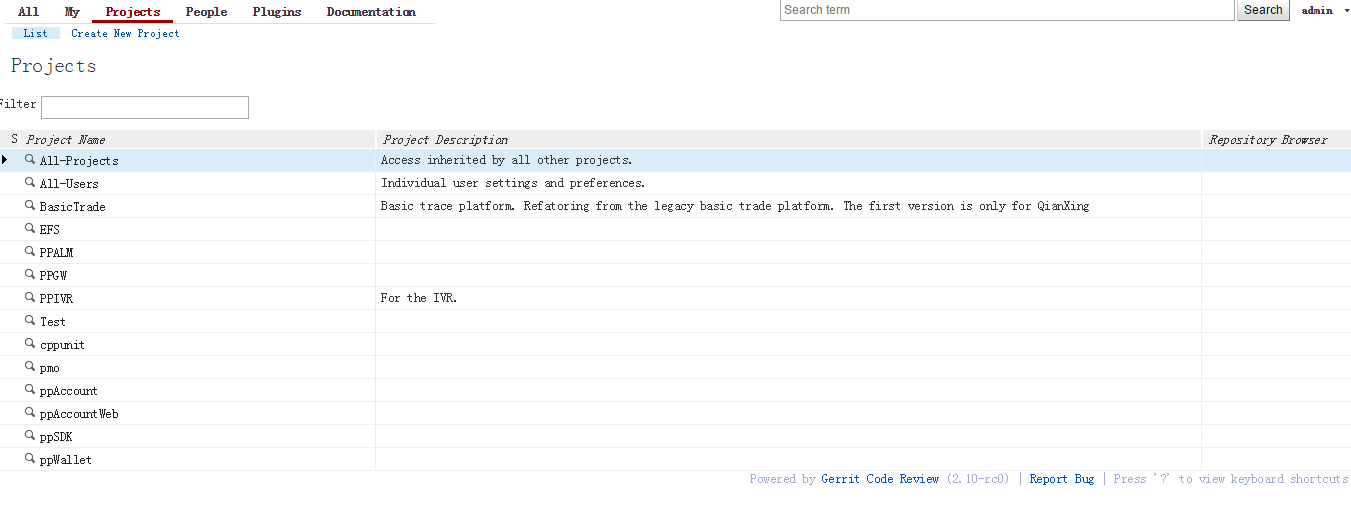


橘黄色Restore为还原选项

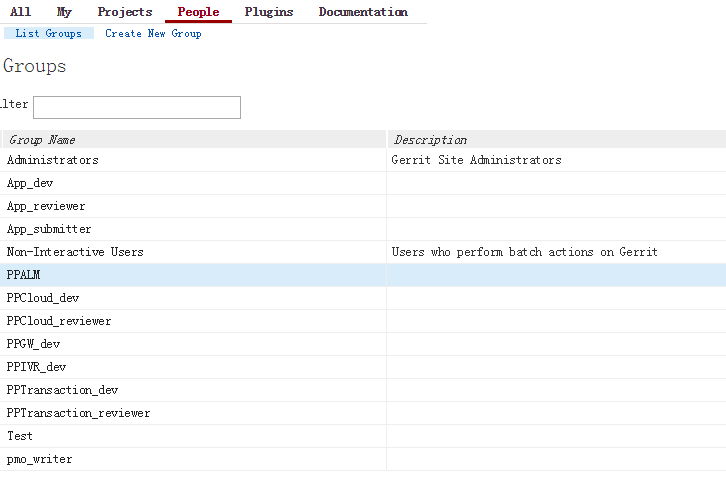




4、gerrit》Projects》List状态表示服务器端所有项目列表

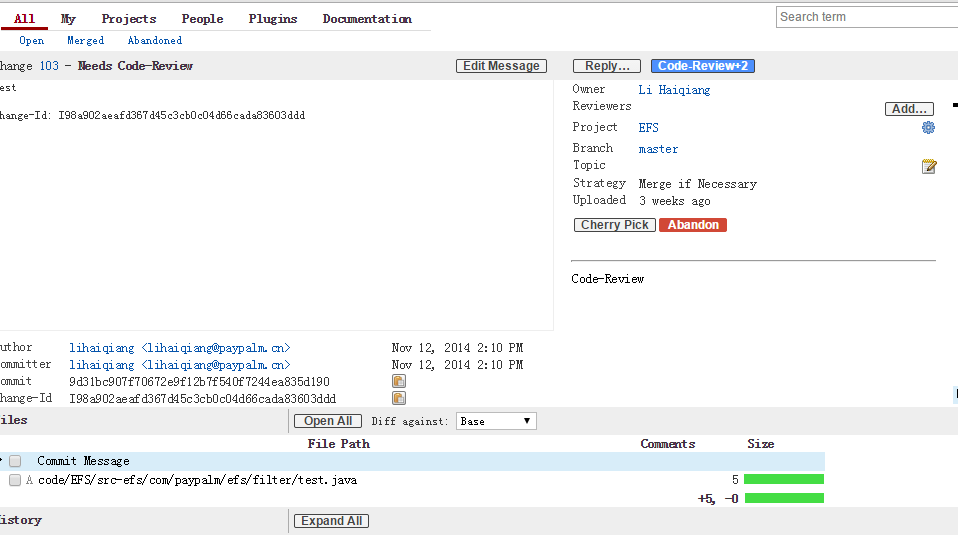


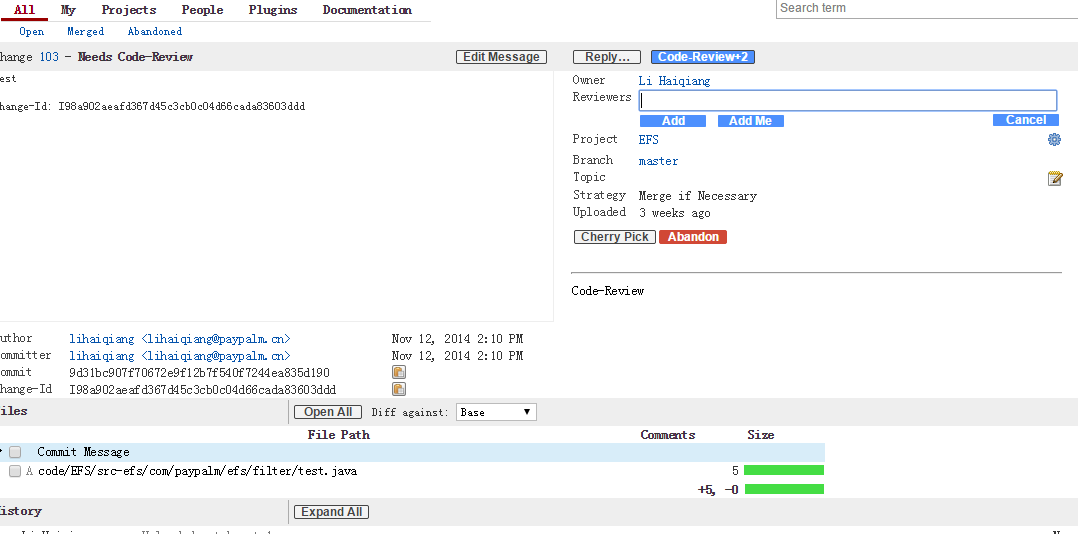
5、gerrit》People》List Groups状态表示所有组列表



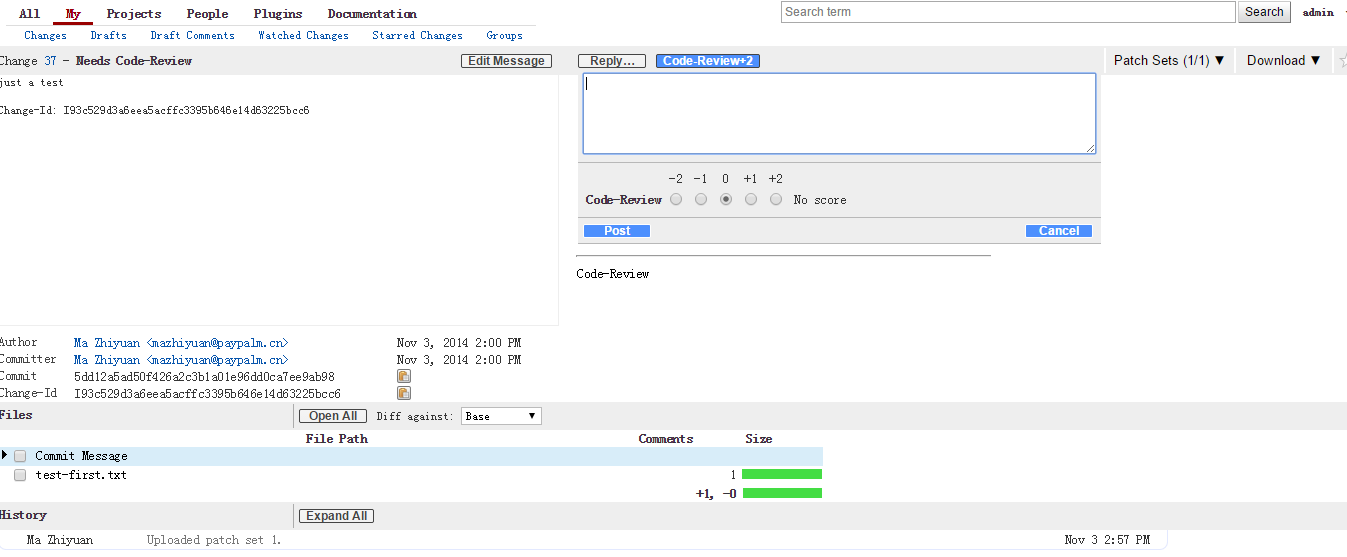
#### 2、评审功能

1、点击进入待评审的patch，点击add添加Reviews人员进行评审

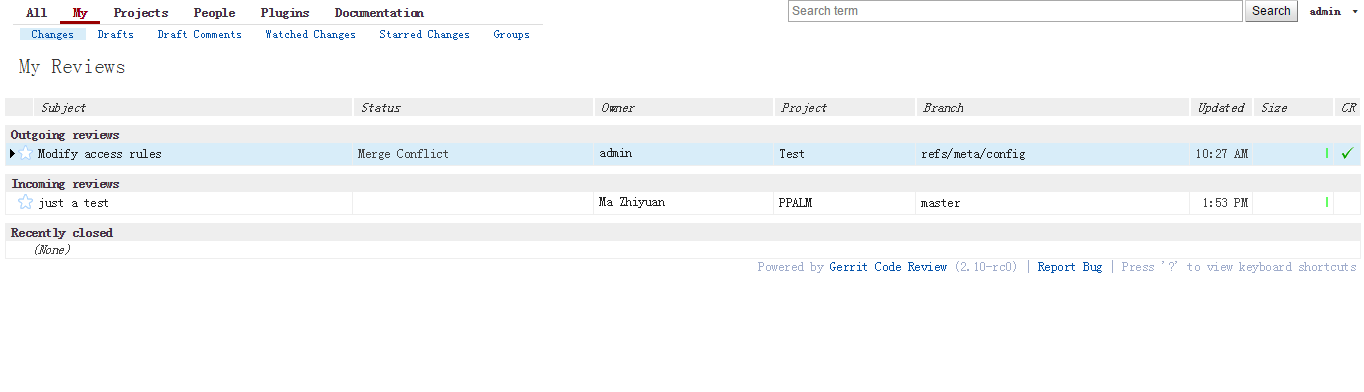




评审人员点击Reply进行评审打分，每一个需要入库的patch必须具备两分+2方可，1分表示自己同意 ，2分 表示完全同意，负分表示不支持此代码入库



2、gerrit》My》Changes状态为需要自己给别人进行评审的状态



### 4、git命令使用

#### 1、账户名和邮箱设置查看

1）、每一个工作环境首先配置在gerrit中注册的账户名和邮箱，请确保一致

git config –global user.name “your-account”

git config –global user.email “your-email”

git config -l

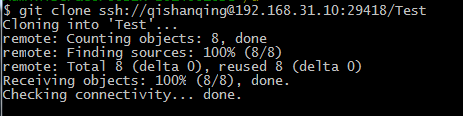
#### 

#### 2、项目库clone

根据gerrit项目列表，查看项目下载地址，选择clone with commit-msg hook&&ssh选项,请确保正确方式进行项目库下载

#### 

git clone ssh://your-accout @192.168.31.10:29418/Test

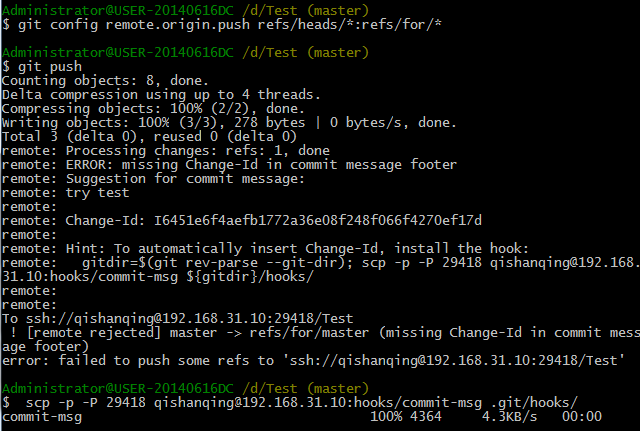


#### 3、提交注意事项

每一个新clone的库第一次提交都需要执行以下步骤(下载服务端钩子到本地库，以便提交评审形成chang-id)

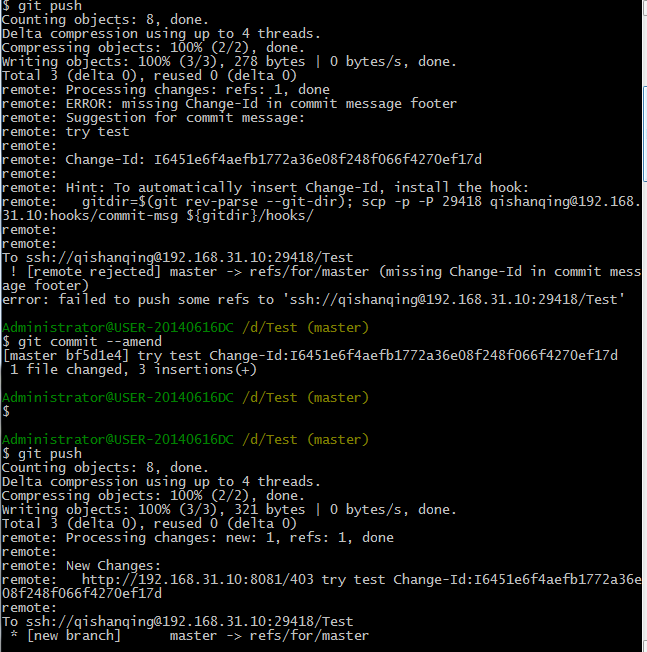
scp -p -P 29418 your-account-name @192.168.31.10:hooks/commit-msg .git/hooks/

git config remote.origin.push refs/heads/\*:refs/for/\*



当执行完以上步骤，第一次git push依然会产生missing Change-Id错误，

用git commit --amend命令把错误信息中的changed id进行添加，如下图

本地工作库中，以最后一次成功push为节点，如果超过两条commit信息也会产生此错误

合并多条commit为一条记录，可以用git reset 后跟要回退到最新push成功的版本号，整合多条记录为一条

如产生uppack error 和 changed closed，建议保存工作库中修改文件，并进行强制回退、重新同步最新代码，以修复工作库index。

git reset –hard xxx

##### 同步项目代码的时候会产生冲突

首先可以先隐藏本地修改的

git stash

同步代码

git pull

恢复刚被隐藏的修改

git stash pop

有冲突进行修改编辑，并进行添加（add）到暂存区后，进行项目开发

#### 4、git具体命令解析

从远端服务器获取本地分支最新版本，任何时候进行开发，请确保工作库为最新状态，这样才能避免与其他开发人员进行冲突

git pull

添加文件或修改文件后将文件添加到暂存区

git add file-name

删除文件后将文件添加到暂存区（整个文件夹删除添加-r参数）

git rm file-name

提交所有暂存区修改到本地库,必须添加提交日志作为以后参考

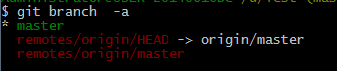
git commit -m “commit massage”

补提交（当一个patch已经push到gerrit但还没有merged，可以进行不提交修改commit massage信息）

git commit –amend 进行编辑

查看当前本地和服务端所有分支，\*号代表当前工作分支

git branch –a



随时查看工作库状态

git status（添加参数 –s为精简模式）

修改后差异输出

git diff

提交日志随时查询

git log

将本地最新分支推送到gerrit准备评审

git push origin HEAD:refs/for/xxx （xxx代表分支名）

清理工作库空间，删除不受控的所有文件

git clean –df

基于当前分支的本地最新状态新建并切换本地分支

git checkout -b new\_branch

合并分支(将target\_branch合并到本地)：

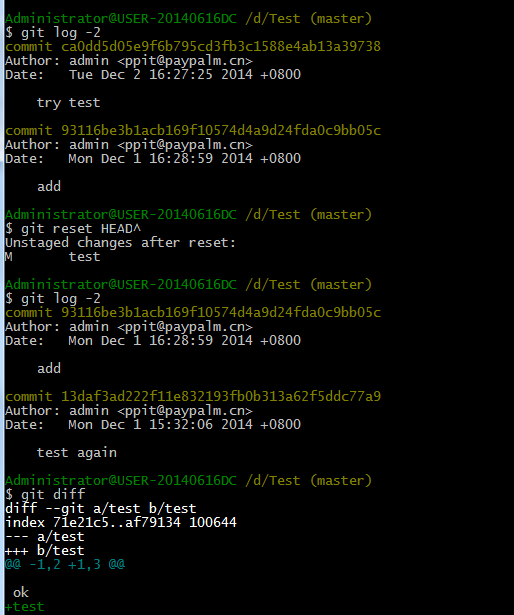
git merge target\_branch

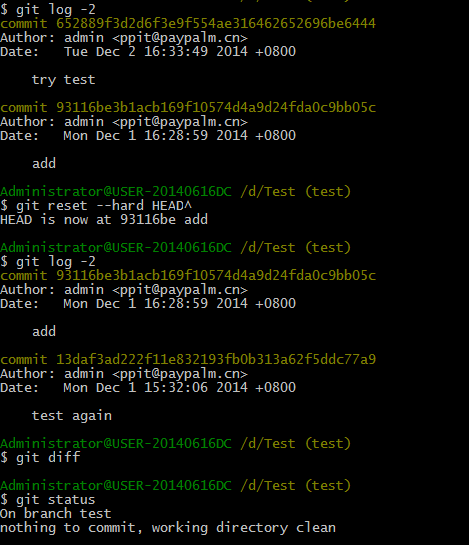
重整分支版本或合并多条提交记录为一条

git rebase base\_branch

重置命令是git最常用的命令，也是最危险最容易误用的命令，使用如下图

git reset xxx （回退log至某个commit值，但是期间的修改得到保留，增加参数—hard回退所有记录和修改文件）





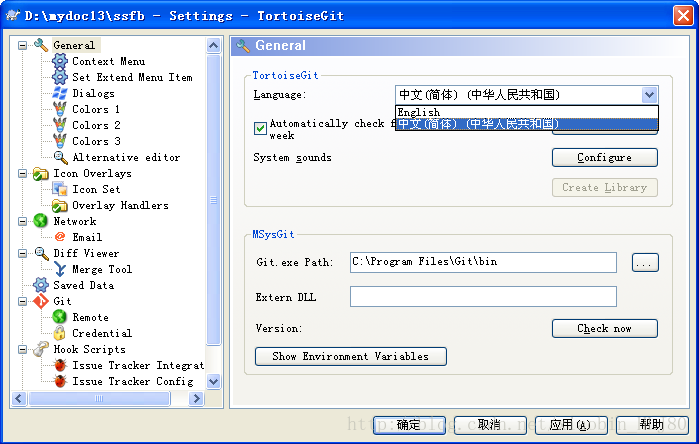
### 5、TortoiseGit客户端使用简介

从<ftp://192.168.31.10/Software/Tool/Git/>

（用户名/密码 paypalm/paypalms）获取软件TortoiseGit安装包和汉化包

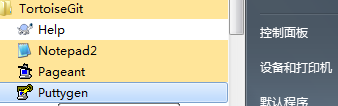
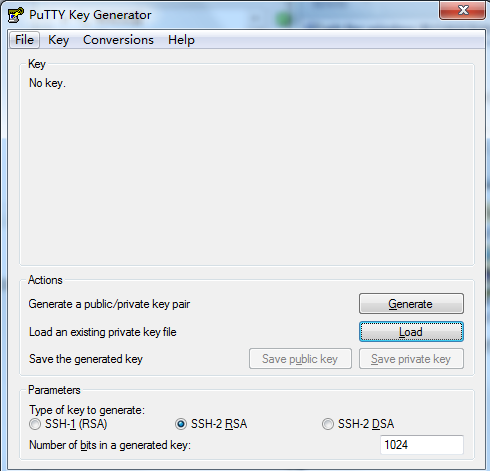
#### 1、安装设置

默认安装后，使用跟svn客户端相类似，在本地点击右键TortoiseGit》setting

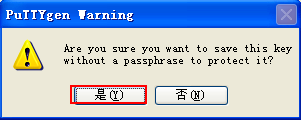
设置为中文界面[](http://files.jb51.net/file_images/article/201409/2014092116554743.png)

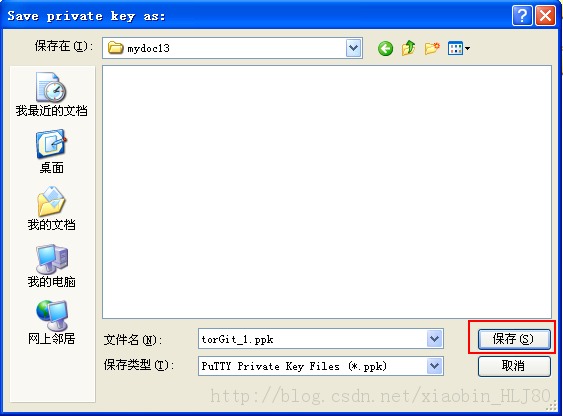
修改git网络设置的ssh客户端链接为下图，不然以后会git push 会失败

#### 

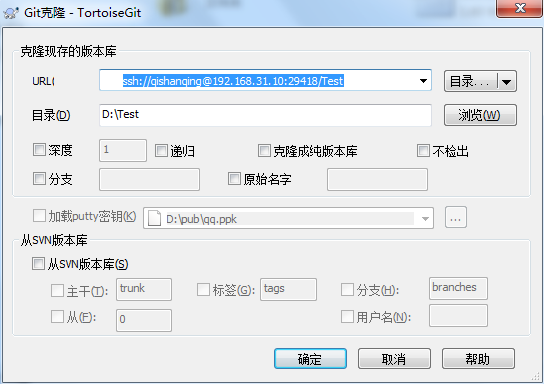
然后用的puttygen进行私钥的格式转换，也可以用工具直接打开，选择load进行加载私钥，成功后save private key，

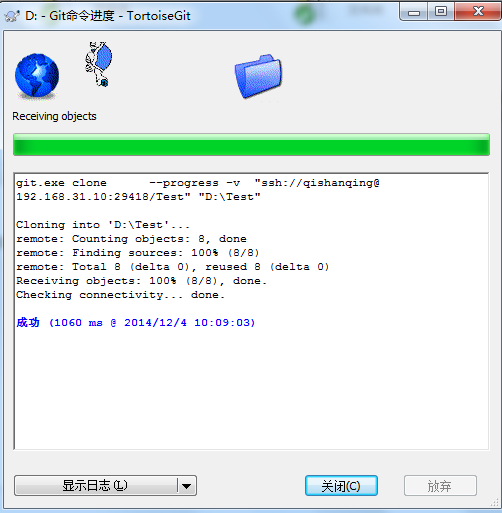
确定生成。





#### 2、Git clone项目库





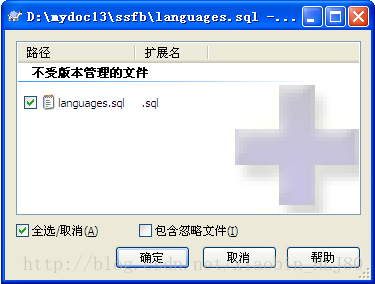
#### 3. 提交本地版本库

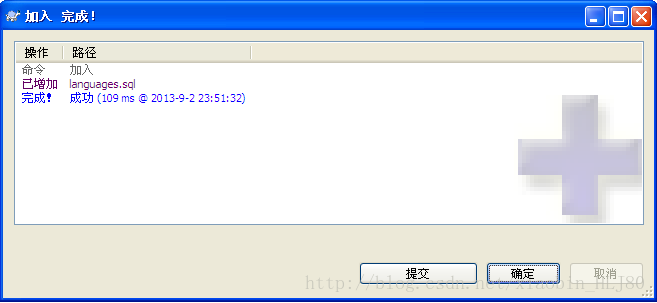
在本地版本库中新建立的文件图标均为http://files.jb51.net/file_images/article/201409/2014092116554861.png

##### 1、 添加到本地库

鼠标右键点击如上图图标的文件，选择“增加”



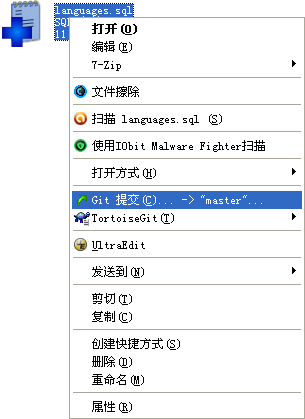


[](http://files.jb51.net/file_images/article/201409/2014092116554964.png)

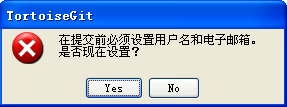
这时，文件图标变为http://files.jb51.net/file_images/article/201409/2014092116554965.png表示添加到本地库成功。

##### 2、 提交到本地库

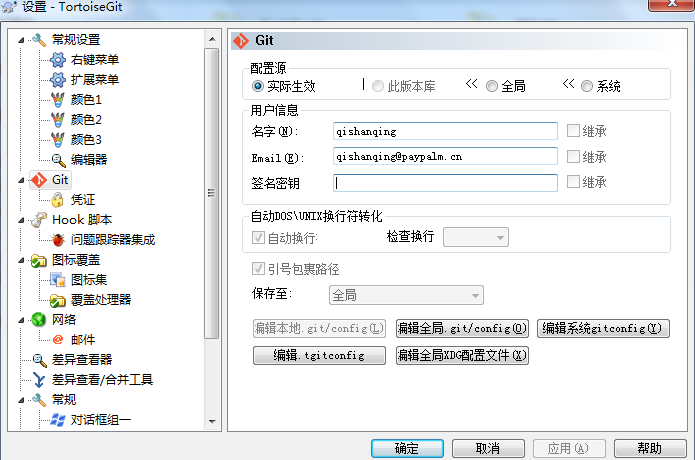
鼠标右键点击如上图图标的文件，选择“Git提交(C) ... -> "master"...”



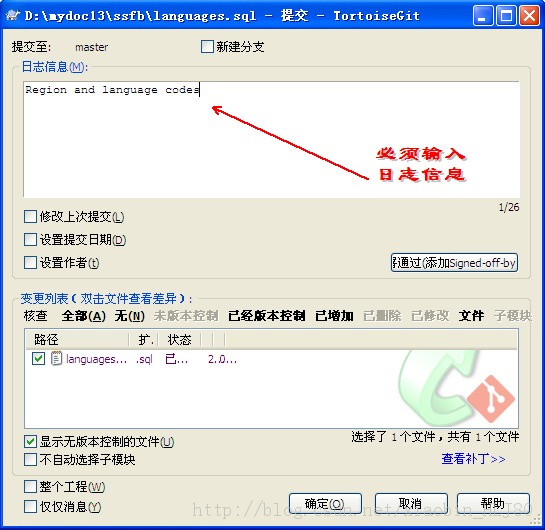
会出现提示，设置用户名和邮箱。

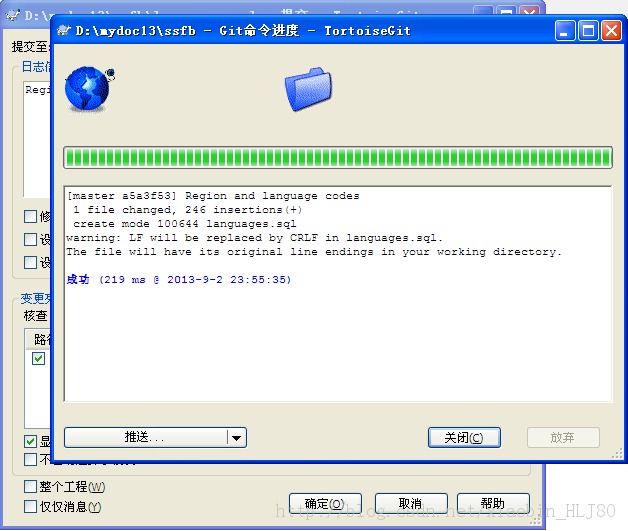


##### 设置用户名、邮箱



##### 输入日志信息，完成提交



[](http://files.jb51.net/file_images/article/201409/2014092116555070.png)

#### 4. 推送本地库到服务器

如果所有文件夹和文件的图标都如图所示http://files.jb51.net/file_images/article/201409/2014092116555071.png，那么我们就可以把本地库推送到版本服务器了。

在本地库所在根目录（本机为D:\mydoc13\ssfc），点击鼠标右键，选择“TortoiseGit”-〉“推送”

