**T10 使用Gerrit进行代码管理使用说明**

**V1.0---20180103**

Gerrit用来对项目代码进行审核管理，所有开发人员提交的代码，必须经过代码审核（review）后，才正式合入到Git库中。

目前在10.10.134.54上建立了gerrit系统，T10项目代码会完全使用gerrit来进行代码和人员权限管理。

T10我们使用ebent10分支作为开发分支，开发人员基于此分支进行开发和提交，目前由管理员将代码合入主分支并与BYD分支合并。

每个开发人员的账户已经建立，如果想更改浏览器登录密码，请联系管理员。如果有人员变动，需要通知管理员进行增删。

Gerrit访问地址： <http://10.10.134.54/gerrit/>

提示：最后的”/ ”必须有。

用户操作：

1. SSH key生成并上传

Gerri下的Git库可使用http或者ssh方式访问代码。Git使用git clone命令时，后面的url如果是[http://\*\*\*\*/T10.git](http://****/T10.git)，那么就是http模式访问，如果用ssh://\*\*\*\*/T10.git，说明使用ssh模式。

SSH模式----推荐使用

SSH模式使用自己生产的key来保证账户自动登录。每个开发人员自己生成一对key文件，pub文件为公钥，其内容需要复制到git服务器上。

举例如下： ---最好加上-C写明key的用途，-f参数后写自己的key文件名

ssh-keygen -t rsa -C 'gerrit宋岩' -f gerritsongyan

连续回车生成一对key文件，私钥放入个人.ssh目录中

mv gerritsongyan ~/.ssh

vim ~/.ssh/config

Host 10.10.134.54

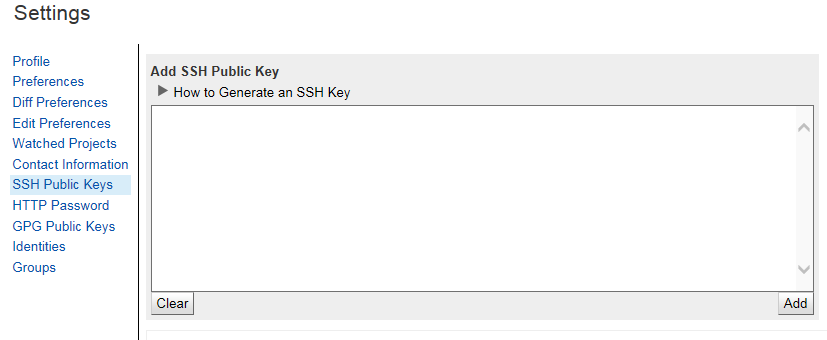
HostName 10.10.134.54

IdentityFile ~/.ssh/gerritsongyan

chmod 600 ~/.ssh/ gerritsongyan

cat gerritsongyan.pub

复制文件内容，粘贴到gerrit页面中（Settings-- SSH Public Keys）



加入key后，用户可以免密码通过ssh协议访问gerrit和git 。

HTTP方式：

每个用户浏览器登录gerrit，Settings中选择“HTTP Password”，点击“Generate Password”，会自动生成一个随机密码，复制下来即可。----注意，刷新页面后看不到原密码，如果忘记，只能再次生成。

如果想使用固定的http密码，联系管理员添加。

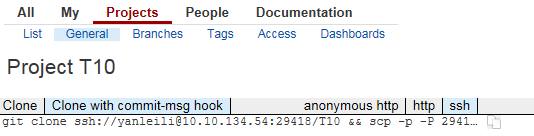
HTTP方式每次pull或者push需要输入用户名密码，可以在clone时把用户名和密码写到url中（不推荐），比如 git clone <http://yanleili:passwd@10.10.134.54/T10.git> ，以后就不需要输入密码，不过缺点是git 配置文件中此url明文保存。

1. 从Git获取代码并设置git参数
2. 第一次获取：

git clone ssh://songyan@10.10.134.54:29418/T10 –b ebent10 && scp -p -P 29418 [songyan@10.10.134.54:hooks/commit-msg T10/.git/hooks/](mailto:songyan@10.10.134.54:hooks/commit-msg%20T10/.git/hooks/)

必须加上后面下载hook脚本的命令，因为gerrit需要Change-Id作为提交的唯一标识，没有hook脚步无法生成ID。

每个人在gerrit的页面可以自行获取具体命令（Projects—List---项目名称—General—Clone with commit--msg hook--ssh）



Clone时加-b ebent10会自动指定当前分支为ebent10，如果直接使用gerrit提供的命令，默认分支为t10\_LA1.1.1\_dev，在clone完成后，需要切换分支：

Git checkout ebent10

1. 利用旧git代码库

如果以前已经有本地T10 git库（原来git远端指向252），那么需要改变git remote指向即可继续使用旧目录，无需重新clone。

例如：

git remote set-url origin ssh://songyan@10.10.134.54:29418/T10

或者直接编辑.git/config文件修改remote url即可。

1. 设置user参数

之前因为需要向BYD提交，所有开发人员的user全部设置为liweichao，现在需要指定为自己的邮箱（gerrit个人settings中已由管理员设置好）

为避免搞混，必须设置本地git库的local 参数

例如：

git config --local user.name "songyan"

git config --local user.email [songyan@eben.cn](mailto:songyan@eben.cn)

1. 从远端更新本地git代码---pull（推荐fetch+merge）

本地git目录下，pull操作从远端获取最新项目代码

git pull origin ebent10

如果本地和远端有log信息不匹配的时候，git会自动进行一次merge并记入log，但是这次merge没有Change-ID，提交时会报错，所以需要进行rebase

git rebase

git会根据log自动比对提交版本并处理掉这次merge信息

之前ebent10里包含的merge信息不需要考虑，只要防止今后使用中产生的。

1. 提交代码

代码更改后，需要正常的git提交操作步骤，add---commit---push

git add \*\*\*.java

git commit –m ‘fix bug \*\*\*’

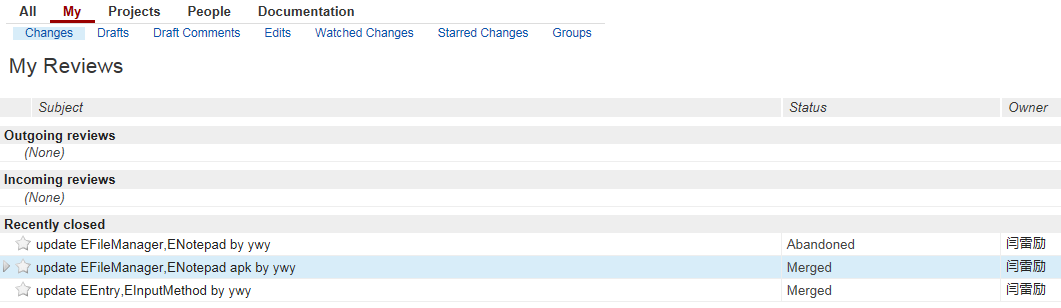
git push origin HEAD:refs/for/ebent10

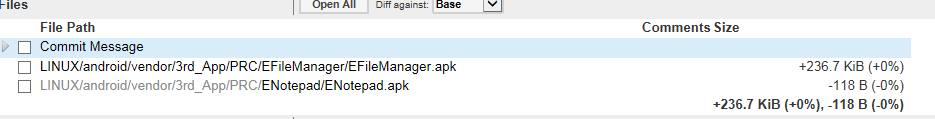
跟之前的习惯操作区别的是push时不能直接push到远端ebent10分支，管理员设置了权限，代码必须经过review才能入库，所以必须是refs/for/ebent10这个分支（gerrit规定refs/for/分支名为review对应分支）

Push完成后，必须在浏览器进行代码review。

具体review操作如下：

My---Changes，看到提交记录，点击进入

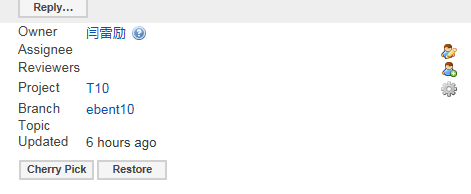


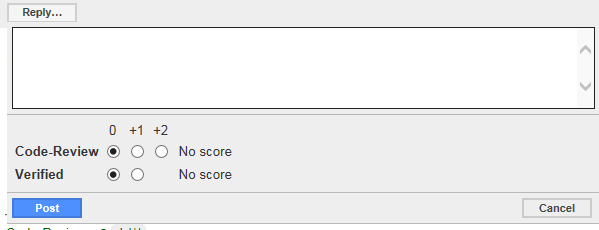


可以看到具体的提交文件，点击可以自动对比并显示文件更改

同一页面下，可以进行review

点击“Reply…”---写代码审核注释，Code-Review +2, Verified +1，然后post





最后点击“Submit”提交，代码会被真正合入ebent10分支，其他人pull时才可以看到。

其他功能，随着使用会陆续加入或者调整。