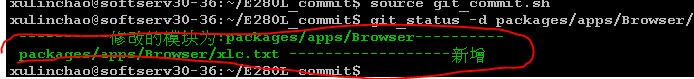
在根目录运行，，或者，然后就可以在根目录使用git\_status,和git\_add指令

git\_status指令是为了得到我们要提交的在根目录修改的文件列表，如下，我在package/apps/Brower/目录下增加个xlc.txt文件，此时在根目录执行



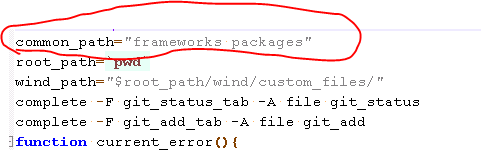
就可以得到如下提示



git\_status 有三个选项，-a，-d,- -help,



-a选项是在脚本配置的路径基础上，增加自己想要查询的模块，



Common\_path就是-a选项配置的默认路径，搜索全面，假如我们不记得修改了哪些模块的代码，可以使用git\_status –a 加上你想要的路径，这个函数的缺点是比较慢，要搜索大量git仓库，然后对比。

如果我们只是修改某个模块，建议使用git\_status –d 加上你想要的模块，效率快。

当我们得到了我们的提交列表时，此时我们可以使用.

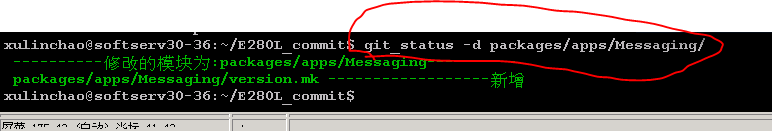
至于git\_status –help是使用帮助，可以查看-a,-d的具体用法

git\_add指令，是提交，如果我们修改的文件时根目录本来就有的代码，但是wind目录没有，按照以前情况，我们是将这个文件复制到根目录下，这个我们没办法通过gerrit去清楚的看见我们的修改，git\_add就是解决这种情况，git\_add会将跟目下有的，而wind没有的文件，首先在wind下做一次干净的提交，自动提交为just add，然后，将我们的修改复制到根目录去，由我们手动提交。（这里git\_add有 –d选项，可以查看git\_add –help，但是不推荐使用，直接使用git\_add）

下面举个例子，

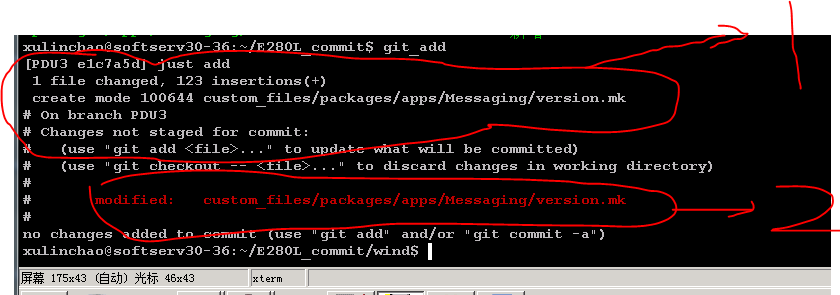
我在packages/apps/Messageing 目录修改文件version.mk，此文件wind目录下没有，

当我用



出现上述提示情况，

此时我只需要git\_add

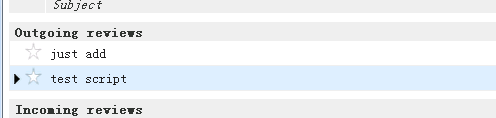


看提示1，是做了次干净的提交，2是将我们修改复制到wind下，

我们将修改提交修改下，提交到gerrit上看下



Gerrit情况



just add是干净的源码

test script是我们修改的，所以我们查看就可以清晰看出我们修改，这样对review或者merge的人能够更好的管控代码质量。

