Gerrit commands:

-------------------------------------------------------------------------

git checkout -b <branch-name>

git-add <file-to-add>

First push:

git-commit -m "iktoolsmmcp-8\_zch45\_a22512\_progresstest"

git push ssh://gerrit.pcs.mot.com:29418/tools\_mmcp/mmcp\_mzf iktoolmmcp-8\_zch45\_a22512:refs/for/master

git push origin <branch-name>:refs/for/master

Rework:

git-commit –-amend <file-to-update

git push ssh://a22512@gerrit.pcs.mot.com:29418/tools\_mmcp/mmcp\_mzf iktoolmmcp-8\_zch45\_a22512:refs/changes/<change\_id>

-------------------------------------------------------------------------

Merge

git checkout <branch-be-merged-to>

git merge <branch-be-merged-from> // this will merge from <branch-be-merged-from> to <branch-be-merged-to>

make changes and commit multiple times on one dev branch,

when you get ready to push to gerrit, we can do like below:

git checkout master -> switch to master first

git checkout -b <new\_branch\_ready\_for\_gerrit\_push> -> create one new branch from the master

git merge --squash <from\_dev\_branch\_which\_contains\_multiple\_commits> -> this will create a commit based on new branch w/o commit

git commit -m "comments" -> commit the new branch with all changes from the dev branch

-------------------------------------------------------------------------

updated Comment on existing commit

git commit --amend -m "New commit message"

git reset --soft HEAD^ //Undo last Git commit

// roll back last wrong commit

git reset --hard <commit\_id> // commit\_id is the last one you want to roll back to

HEAD 最近一个提交

HEAD^ 上一次

git push origin HEAD --force

-------------------------------------------------------------------------

Update tag:

git tag MTK-SDK\_N\_12\_26\_01I\_ZCH45 -m "MTK-SDK\_N\_12\_26\_01I\_ZCH45"

git push ssh://gerrit.pcs.mot.com:29418/tools\_mmcp/mmcp\_mtk\_sdk.git master MTK-SDK\_N\_12\_26\_01I\_ZCH45

====== Make sure changes are sync to LV server first then continue this

git push ssh://a22512@il93lnxgit01.pcs.mot.com/proj/repo01/tools/platform\_tools/dev/mmcp\_mtk\_sdk.git master

git push ssh://a22512@il93lnxgit01.pcs.mot.com/proj/repo01/tools/platform\_tools/dev/mmcp\_mtk\_sdk.git master MTK-SDK\_N\_12\_26\_01I\_ZCH45

-------------------------------------------------------------------------

remote sources:

git clone ssh://a22512@il93lnxgit01.pcs.mot.com/proj/repo01/tools/platform\_tools/dev/

git clone /proj/repo01/tools/platform\_tools/dev/

--> new

[pull] ssh://il93lnxgit01.pcs.mot.com:29418/tools/platform\_tools/dev/

[push] ssh://gerrit.pcs.mot.com:29418/tools/platform\_tools/dev/

git clone ssh://zch45app06.sc.mcel.mot.com:29418/tools\_mmcp/mmcp\_\*

git clone ssh://gerrit.pcs.mot.com:29418/tools\_mmcp/mmcp\_\*

git clone ssh://zch68rep01.ap.mot.com:29418/tools\_mmcp/mmcp\_

/proj/repo01/gerrit/tools\_mmcp

-------------------------------------------------------------------------

git push origin master # 提取我的master分支 并更新到远程仓库的master分支

git push origin serverfix # 提取我的serverfix分支 并更新到远程仓库的serverfix分支

git push origin serverfix:remoteserverfix # 提取我的 serverfix分支 并更新到远程仓库的remoteserverfix分支

git push origin :serverfix # 删除远程分支 serverfix（在这里提取空白然后把它变成[远程分支]）

-------------------------------------------------------------------------

Windows GIT msysgit

How to remove troubling "Local uncommitted changes, ...."

There are sevearal reasons may cause this:

1) file mode check "old mode 100755 new mode 100644"

It's due to we clone the repo from UNIX and defaultly admin will set the filemode to true for evey repo

After we copy them from unix machine to Windows machine, due to "old mode 100755 new mode 100644" problem, Git on Windows will report this change though we didn't change anything.

To avoid this, "git config --add core.filemode false"

2)

git-add

git-rm

git-mv

git-commit

git

compare:

git status: show the all the status (index-current HEAD commit; working tree-index; working tree not tracked by git)

git diff: diff working tree against index

git diff --cached: diff index against local repro (latest commit)

git diff HEAD: diff working tree against latest commit

git log: show commit log

git checkout -b branchname tag

$ git reset –-hard HEAD //–hard表示将working tree和index file都撤销到merge以前状态

push and delete remote branches

This is an action that many Git users need to do frequently, but many (including the author) have forgotten how to do so or simply don’t know how. Here’s the definitive guide if you’ve forgotten.

So let’s say you have checked out a new branch, committed some awesome changes, but now you need to share this branch though with another developer. You can push the branch up to a remote very simply:

git push origin newfeature

Where origin is your remote name and newfeature is the name of the branch you want to push up.

This is by far the easiest way, but there’s another way if you want a different option.

Geoff Lane has created a great tutorial which goes over how to push a ref to a remote repository, fetch any updates, and then start tracking the branch.

Deleting is also a pretty simple task (despite it feeling a bit kludgy):

git push origin :newfeature

That will delete the newfeature branch on the origin remote, but you’ll still need to delete the branch locally with git branch -d newfeature.

There’s a script called git-publish-branch created by William Morgan that can easily automate this process if you find yourself performing these actions frequently. It also makes deleting remote branches feel a bit more natural. Know of better or easier ways to do the above tasks? Let us know in the comments or submit your own tip!

git-checkout -b <branch\_name> #create and checkout to the branch

git-branch -D <branch\_name> # delete a local branch ANYWAY

git-commit -m <comment>

git-checkout master

git-merge <branch>

git push origin master

#################################################################################################################3

Hi Wang Chen

We found you still sync code from gerrit server , pls use the following command sync :

For RSD common repos:

git clone /proj/repo11/tools/platform\_tools/dev/hdt\_rsd\_rsdl.git

git clone /proj/repo11/tools/platform\_tools/dev/hdt\_css\_common.git

.....

For RSD mmcp repos:

git clone /proj/repo01/gerrit/tools\_mmcp/mmcp\_oflash.git

git clone /proj/repo01/gerrit/tools\_mmcp/mmcp\_opst.git

......

For Bootloader:

change the ~/.gitconfig file as below:

[url "/proj/repo01/gerrit/domino"]

insteadof = "ssh://gerrit.pcs.mot.com:29418/domino"

then use the below command to get the code from NJ local server:

a.domino:

git clone /proj/repo01/gerrit/domino/engine\_qcom.git

git clone /proj/repo01/gerrit/domino/build\_tools.git

git clone /proj/repo01/gerrit/domino/engine\_boot.git

git clone /proj/repo01/gerrit/domino/qcom\_signing\_tools.git

git clone /proj/repo01/gerrit/domino/qcom\_signing\_tools.git

git clone /proj/repo01/gerrit/domino/engine\_pupd.git

git clone /proj/repo01/gerrit/domino/engine\_security.git

b.avon

git clone /proj/repo01/gerrit/avon/atso\_tools.git

git clone /proj/repo01/gerrit/avon/engine\_boot.git

git clone /proj/repo01/gerrit/avon/engine\_qcom\_wcdma.git

How to ignore some files in GIT repo:=============================================================

If you want to ignore certain files when checking in, add them in .ignore file. I attached it here. This file ignores .obj, .pch, etc.

If it is already indexed your ignore file won't work. You need to call git rm first to remove the files from index:

git rm --cached \*.plg,

git rm --cached \*.obj

git rm --cached \*.sbr

etc to remove from index.

After doing that, .ignore file should work. git status will not display the ignored files.

Wulei's sample:

git push ssh://ctf874@gerrit.pcs.mot.com:29418/tools\_mmcp/mmcp\_opst iktoolsmmcp-171\_zch45\_ctf874\_opst:refs/for/master

How to tag:

create remote tag:

git tag -m "<comments>" <tag-name>

git push origin <tag-name>

sample:

git tag -m " for nc release" tag1

git push origin tag1

git tag -m "INT\_MTKCLIENT\_N\_12\_15\_01I\_ZCH45" INT\_MTKCLIENT\_N\_12\_15\_01I\_ZCH45

git push ssh://gerrit.pcs.mot.com:29418/tools\_mmcp/mmcp\_mtk\_client.git INT\_MTKCLIENT\_N\_12\_15\_01I\_ZCH45

git push ssh://ghc837@il93lnxgit01.pcs.mot.com/proj/repo01/tools/platform\_tools/dev/mmcp\_mtk\_client.git INT\_MTKCLIENT\_N\_12\_15\_01I\_ZCH45