

Create new feature branch

```
git checkout master
git fetch --all
git pull --rebase
git checkout -b featur e/####
```

Always create feature, fix or hotfix branch from master.

Merge into dev or any enviroment branch

```
git checkout dev
git fetch --all
git pull --rebase
git merge --no-c ommit --no-ff featur e/####
if have merge conflicts resolve them with any tool. Theb:
git commit -m " Merge featur e/### into Dev"
git push origin dev
```

*--no-c ommit Indicates don't create an imediate commit.

*--no-ff No fast Fordwards. This keep yout repo history cleaner

Delete branches already on master

```
git branch -r --merged master
git branch -r | egrep " (*| bra nch 1|b ran ch2 )" | sed 's/ori gin \\\\/' | xargs -n 1 git push origin --
delete
```

*-r includes all remote branches

"(|branch1|branch2)" include all branches for first command.

Pull master changes into working branch

```
git checkout master
git fetch --all
git pull --rebase
git checkout workin gbranch
git rebase master
if have merge conflicts resolve them with any tool.
Then:
git rebase --continue
else
git push origin workin gbranch
```

Create RC Branch

```
git checkout master
git fetch --all
git pull --rebase
git checkout -b RC/1.0.0.#
/ Merge or PR all feature branches for the release /
```



Reset branch

```
git checkout working branch
git reset --hard
git push origin working branch
```

Recomendations

Can i merge dev into master?

No

Can i merge two features?

It's not recommended, but if this is a must, you can create a third branch, which contains both features

How often must pull master changes?

At least once we weak must pull master changes to all branches

