Contents

[Clone an existing repository 2](#_Toc423040709)

[Working on an existing branch. 2](#_Toc423040710)

[Creating a new branch. 3](#_Toc423040711)

[Check-in your changes 3](#_Toc423040712)

[Checking out someone else’s branch 3](#_Toc423040713)

[Updating Remotes 3](#_Toc423040714)

[When someone else has updated your branch 4](#_Toc423040715)

[Approach -1 4](#_Toc423040716)

[Approach -2 4](#_Toc423040717)

[Taking a backup of your branch 4](#_Toc423040718)

[Saving your un-committed changes temporarily 4](#_Toc423040719)

[Rebasing 4](#_Toc423040720)

[Merging 5](#_Toc423040721)

[Pull Requests 5](#_Toc423040722)

[Branch Commands 5](#_Toc423040723)

[🡪Delete a Local branch 5](#_Toc423040724)

[🡪Delete a remote branch 5](#_Toc423040725)

[🡪Rename a branch locally 5](#_Toc423040726)

[🡪Get latest from remote branch – [Merge with local] 5](#_Toc423040727)

[🡪Get latest from remote branch – Overwrite local 6](#_Toc423040728)

[🡪Create a local tracking branch of Existing Remote Branch 6](#_Toc423040729)

[🡪Pushing New Local Branch Upstream 6](#_Toc423040730)

[🡪Show all branches 6](#_Toc423040731)

\*All commands to be run at the Git-Bash command prompt.

\*Do not use Visual Studio for any communication with the GIT repo.

\*Create a new branch for each new feature or common component.

\*Write meaningful commit messages.

Clone an existing repository.

git clone <http://url-to-repo.git>

**When to use:** Use this when you want to download an entire solution from an existing remote repository in your current development environment in your local project folder. This is typically done when you are starting new in the project. You should not be doing this once your git repo is initialized from a previous clone.

## Working on an existing branch.

git fetch --all : This will update all remote branch reference points.

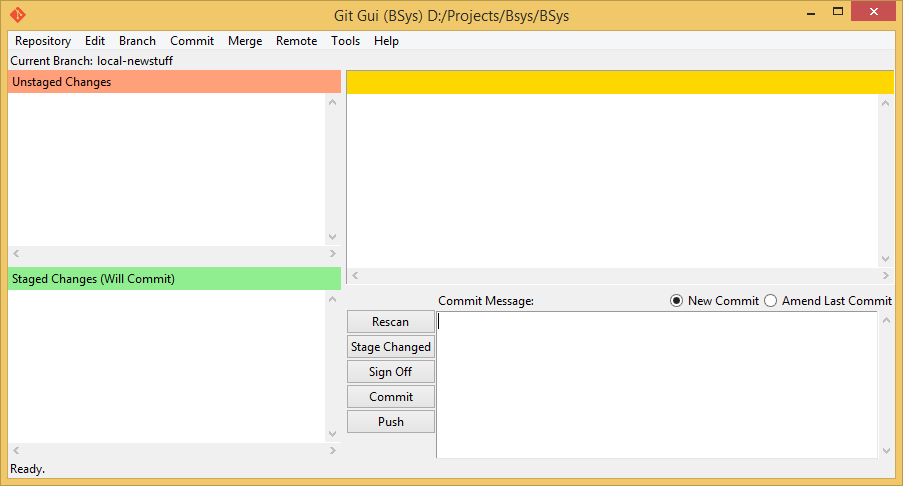
git branch –avv : Show a list of existing branches and their remote mappings. Ensure that the branch you are going to create locally doesn’t already exist on your local repository.

git checkout –b local-dev origin/dev : This creates a branch called local-dev that maps to the remote branch dev. Any changes committed and pushed to this local branch gets mapped to the server branch dev. This mapping will get uploaded to the remote when the branch is pushed.

git commit –am ‘commit message here’: This will update the local repository with your changes.

Alternatively use the git gui. Type the following in the command line:

git gui &



* All files modified will appear in the un-staged changes section.
* Click on each file icon to move them to the staged changes section.
* Enter the commit message and click on [Commit].

git push origin: This will update the remote repository to which the local branch is mapped with the changes in the local repo.

**When to use:** Use thiswhen there is an existing branch in the remote server that you want to track. Tracking here means that every change you make can be pushed upstream to the remote branch without explicitly having to set that up.

## Creating a new branch.

git branch –avv : Show a list of existing branches and their remote mappings. Ensure that the branch you are going to create locally doesn’t already exist on your local repository.

git checkout -b branch-name : This creates a branch locally that is not yet sent up to the server.

{Use git gui as shown above to commit changes} or the command below.

git commit –am ‘commit message here’: This will update the local repository with your changes.

git push -u origin branch-name: This pushes your newly created local branch to the new remote branch created with the same name. The branch name needs to be the same branch name that you just created locally. It will create a remote branch of the same name as well.

git push -u origin branch-name:another-name : This pushes your newly created local branch branch-name to the new remote branch created with a different name called another-name.

**When to use:** Use this when you are going to upload a local branch that is not there on the remote side. You have typically started out a new feature, creating a new branch in your local repo to be uploaded to a remote repository at a later date.

## Check-in your changes

Refer to the state of the branch you are working on; new branch or existing branch and follow the steps accordingly.

Checking out someone else’s branch.

git checkout origin/branch-to-checkout

**When to use**: For just having a look at another branch and not making any changes that can be synced up to its remote. Refer to the section: {[Working on an existing branch](#_Working_on_an)} if you want to modify it locally.

## Updating Remotes

git fetch –all

**When to use**: This is typically run just before you will pull the latest changes into your branch or when you will create a new local branch off some existing remote branch that is not yet in your local repository.

## When someone else has updated your branch

### Approach -1

git fetch --all

git pull origin branch-name

**When to use:** When you need to maintain your changes and also get the latest of someone else’s push commits in your branch. There will be merge conflicts. You will then need to resolve the merge conflicts one by one using your merge tool and then commit and run the push command.

### Approach -2

git fetch --all

git reset –hard origin/branch-to-overwrite

**When to use:** When you don’t care about your local branch state. You just want to overwrite your local branch with what was on the remote.

## Taking a backup of your branch

git checkout –b my-branch-backup

**How to use:** Move into the branch that you intend to back-up and run the command above.

## Saving your un-committed changes temporarily

git stash : This pushes your local changes into a temporary location.

git stash pop: This brings back your local changes from the temporary location.

**When to use:** When you want to save your local changes and work on some other branch so you can switch branches without committing and polluting your commit history with an unfinished commit.

## Rebasing

git checkout -b <branch name>  
// work  
git commit -am “I did this”  
// work  
git commit -am “I did that"

//You can also use the git gui tool [as mentioned above](#_Working_on_an) for commits

git fetch --all  
git rebase origin/master

//you should now be on top of master

//incase of merge issues, run your mergetool using the command below.

git mergetool :Look up how to configure mergetool online

git push –f origin branch-name : You need to force your updates typically fixes to merge conflicts to the remote repo after a rebase.  
//Create a [pull request](#_Pull_Requests)

//watch for build failures [here](https://build.dev.maggroup.com/project.html?projectId=mrg_MrgApps&tab=projectOverview) after a pull request was created  
//when your PR is +1'd  
git fetch -a  
//verify master hasn't changed  
//wait for green build  
//hit merge button on [GitHub.com](http://github.com/)

**When to use:** When you need to bring in changes from another branch that you earlier branched out from. This is typically done in a situation when you branched out from say branch-A into branch-B. While you were working on branch-B, branch-A had some updates. You now need these updates in your current branch which is branch-B.

## Merging

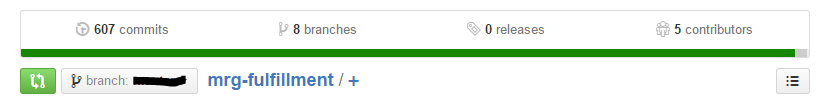
git checkout branch-A

git merge branch-B

**When to use:** When you want to merge the contents of branch-B into branch-A

## Pull Requests

Pull requests are created by clicking the green button on your branch



**When to use:** When you need to submit your code for team review.

## Branch Commands

### 🡪Delete a Local branch

git branch -d the\_local\_branch

### 🡪Delete a remote branch

git push origin :the\_remote\_branch

### 🡪Rename a branch locally

git branch –m new-branch-name

### 🡪Get latest from remote branch – [Merge with local]

git fetch --all

git pull origin branch-name

This will lead to conflicts if someone else worked on the same remote. You will then need to resolve the merge conflicts one by one using your merge tool.

### 🡪Get latest from remote branch – Overwrite local

git fetch --all

git reset –hard origin/branch-to-overwrite

### 🡪Create a local tracking branch of Existing Remote Branch

git fetch --all

git checkout -b localDEV origin/DEV

This creates a branch called localDEV that maps to the remote branch DEV. Any changes committed and pushed to this local branch gets mapped to the server branch DEV

### 🡪Pushing New Local Branch Upstream

git push -u origin feature\_branch\_name

In case you need to push a new local only branch to the server.

For the command above, you currently need to be in the new local branch that you want to push.

### 🡪Show all branches

git branch -avv