# GitHub OverView

<http://butst.usgs.gov/open-source/>

<http://butst.usgs.gov/open-source/gitorious/>

Origin

Commit

Add

On the internet

Access: Laura, Bob

GitHub Remote

Index

Push

Pull

Local computer

Master

GitHub Local

Working Copy

(R Code for example)

## Basic git commands:

## From Windows: GitBash

## From Mac: Terminal

First, change to the correct directory (ex: cd D: cd LADData cd “R Code” cd WRTDS brings you to D:\LADData\R Code\WRTDS)

**Adding a file to internet (orgin):**

git add <filename>

*(this adds the file to the GitHub Local)*

git commit <filename> -m ‘brief message about changes’  
git commit –a

*(this commits all changes)*

If you don’t add a message to the commit, it will send you to Vim.

One way to type a message in VIM:git

Type “i” to get in insert mode (no return)(allows you to type a message). Then hit escape to get out of insert mode. Then hit :wq to ‘write and quit’ (save and quit).

Another way to type a message in VIM:

Type the message in there and then hit Escape ZZ (escape puts Vim in ‘command mode’ ZZ saves and quits – other Vim commands can be found here:

<http://www.tuxfiles.org/linuxhelp/vimcheat.html> )

git push

*(this pushes the commits (the added files in this case) to the origin (internet))*

**Removing a file:**

git rm <filename>

git commit <filename> -m ‘brief message about changes’

git push

**Updating a file from your computer to the internet:**

First, locally update the file/code.

git commit <filename> -m ‘brief message about changes’

git push

**Updating a file from the internet to your computer:**

git fetch vs git pull

git pull = fetch + merge

**Setting up your computer for the first time (cloning what’s on the internet)**

Follow installation instructions here:

<http://help.github.com/win-set-up-git/>

or here:

<http://help.github.com/mac-set-up-git/>

Mac note: I found that I couldn’t see the .ssh folder. To view all folders and files in the Mac Finder, go to the Terminal and type:

defaults write com.apple.finder AppleShowAllFiles TRUE

Then, log out and log back in (the changes will not take place until you do this).

Windows: my ssh key is found here:

C:\Users\ldecicco\.ssh\id\_rsa.pub

The development USGS site is gitorious:

First, change to the directory you want the files to go (cd D: cd LADData cd “R Code” cd WRTDS)

git clone [git@github.com:USGS-CIDA/WRTDS.git](mailto:git@github.com:USGS-CIDA/WRTDS.git) .

The period at the end means put clone that repository where you currently are. Otherwise, you could specify the location.

**Setting up a 2nd computer**

It is OK to set up 2 SSH keys, otherwise copy the following files from your original computer to the new computer:

C:\Users\*username*\.ssh\id\_rsa

C:\Users\*username*\.ssh\id\_rsa.pub

Then, clone the repository as described above.

You might have to set your name and email using this:

git config --global user.name”decicco”

git config --global user.email[”decicco@usgs.gov](mailto:)”

git status  
git rm (this works on the local repository – then commit, then push)  
git mv  
git log  
git blame <file>

========================

git fetch vs git pull  
git pull = fetch + merge  
git push [origin master]

(just type git pull / git push will make the 2 local and remote repositories sync’ed)

If you are trying to type a file or folder with a space, you can either double quote it, or type\ before the space.

**Merge**

If someone commits a change, and you forget to pull, and you also make changes to that file – when you try to commit, git will tell you to resolve the conflicts and merge. Opening up the file, you should be able to see where the conflicts are. Resolve the conflicts, save the file, then to commit, use the following:

git commit -i myfile.R

The -i basically tells it to stage additional files before committing. Another option is:

git commit –a

but this forces you to commit all changes you’ve made to all files, which might not be what you want.

**Uncommit the last change:**

git reset --hard HEAD^

**Branching**

Let’s say Jordan made a fork (a totally different repository), and I want to bring in his changes to my computer – but keeping his work separate. I would create a branch:

git branch matrixTest

Switch to that branch:

git checkout matrixTest

Add the remote:

git remote add jordan url….

Then bring in his work:

git pull jordan master

git branch

Tells you which branch you are currently in

## Gitorious + RStudio

Create a Project:

Project -> New Project

Get some stuff written

Then:

Project -> Project Options->Version Control

Choose git

Then,

**Commit some changes**:

Tools->

To initially set up the connection to gitHub or gitorious:

Tools -> Shell

Then do:

git checkout

git remote add origin [git@butstlnx1.er.usgs.gov:ldecicco/glri-tcl.git](mailto:git@butstlnx1.er.usgs.gov:ldecicco/glri-tcl.git)

\*git push origin master

Try this next time

git push –u origin master (?)

<http://support.rstudio.org/help/discussions/problems/2003-it-seems-that-if-git-remote-was-specified-at-the-command-line-that-its-not-reflected-in-the-rstudio-ui>

I couldn’t push/pull in RStudio until I did this:

git branch --set-upstream master origin/master

## Forks/Cloned Repositories:

In gitorious, I created clone to do my work (a fork in gitHub). Now in my RStudio project, I want to switch my project to pushing to that clone:

git remote rm origin

git remote add origin git@github.com:aplikacjainfo/proj1.git

git config master.remote origin

git config master.merge refs/heads/master

Or….just pull request…

Then for RStudio’s sake:

git push –u origin master

After committing some changes and pushing them to the fork – I can do a Merge Request on gitorious (or Pull Request on gitHub). On gitorious, there’s a nice page that shows what was changed with comments.

I made a new remote (remote = pointer to some git server):

git remote add canonical git@butstlnx1.er.usgs.gov:ldecicco/glri-tcl.git

git remote add ldeciccoGLRI git@....ldeciccos-glri=tcl.git

When you actually want to do the merge:

# Check out a new branch for integration

git checkout -b merge-requests/1

# Fetch the merge request into this branch

git pull git://butstlnx1.er.usgs.gov/ldecicco/glri-tcl.git refs/merge-requests/1

# Show the commits, assess they are okay

git log --pretty=oneline --abbrev-commit master..merge-requests/1

# To apply the changes to your branch:

git checkout master

git merge merge-requests/1

git push origin master

## Setting up a 2nd computer with RStudio + project already created.

At least on the Mac, I had to clone the repository before creating an R project. So, I created a folder in the Finder. Then, I opened up the Terminal and navigated to that folder. Then:

git clone [git@butstlnx1.er.usgs.gov:ldecicco/glri-tcl.git](mailto:git@butstlnx1.er.usgs.gov:ldecicco/glri-tcl.git) .

Now, the files were all there. Then, I opened up RStudio and made a new project in that folder. I was immediately able to pull/push.

## Conflicts in a merge:

If there’s a conflict during the merge request (gitHub: pull request),

<http://weblog.masukomi.org/2008/07/12/handling-and-avoiding-conflicts-in-git>

## Pull from canonical, push to clone (fork):

Make a remote that fetches from the canonical, pushes to the fork, we’ll call it devRemote:

git remote –v

or

git remote show devRemote

that will show push/pull urls

Create a canonical branch to keep clean:

git remote add canonical [git@butstlnx1.er.usgs.gov:ldecicco/dataretrieval.git](mailto:git@butstlnx1.er.usgs.gov:ldecicco/dataretrieval.git)

git checkout –b canonical

git config –l

git config branch.canonical.remote canonical

git push –u canonical master

With this branch…make sure to keep it up-to-date with the master. Then on merge requests, simply do this in the canonical branch:

git checkout canonical  
git pull git://butstlnx1.er.usgs.gov/ldecicco/glri-tcl.git refs/merge-requests/1  
git pus

The refs/merge-requests/1 thing is created in gitorious

Create a development branch that pulls from canonical, pushes to fork:

git remote add devRemote [git@butstlnx1.er.usgs.gov:ldecicco/dataretrieval.git](mailto:git@butstlnx1.er.usgs.gov:ldecicco/dataretrieval.git)

git remote set-url –push [git@butstlnx1.er.usgs.gov:~ldecicco/ldecicco/ldeciccos-dataretrieval.git](mailto:git@butstlnx1.er.usgs.gov:~ldecicco/ldecicco/ldeciccos-dataretrieval.git)

git checkout –b devBranch

git config branch.devBranch.remote devRemote

git push –u devRemote devBranch

Now I develop in the devBranch. When I like what I’ve done, I make a merge request:

‘Request merge’

Be sure to pick the devBranch as the source

If no conflicts:

Switch to canonical branch:

git checkout canonical  
git pull git://butstlnx1.er.usgs.gov/ldecicco/glri-tcl.git refs/merge-requests/1 (or something like this..see the ‘How to apply this merge request’)  
git push

## Check out a new branch from the master

git checkout –b newBranch

git push –u origin newBranch

if it’s a branch for packageBuilds or Documentation, and you want to delete a bunch of stuff, easiest to just do it in the shell (well, delete everything in Windows Explorer or Finder, then go to shell):

git commit –a –m’Deleting files in brang…’