How to use Git and Dropbox together:

This is how I created the enENCOMMAS Git Repo directory: - you must have Dropbox on your machine installed. For a Mac, this puts a directory called /Users/$HOME/Dropbox in your home root directory (here italics mean what you type at the command line):

```
cd ~/Dropbox
mkdir -p repos/enNCOMMAS.git
cd !$
git --bare init
```

This creates a bare directory we can create a master repository into. We now have to create a cleaned version of the enNCOMMAS directory, then create our local git database. First cd into the enNCOMMAS directory. We can recursively remove all of the .svn (or CVS) directories using a quick command inside the enNCOMMAS directory:

```
find . -name '.svn' -type d -print | xargs rm -rf {}
```

or

```
find . -name 'CVS' -type d -print | xargs rm -rf {}
```

Then, we create the local git repository for enNCOMMAS:

```
git init
git add .
git commit
```

You now have create a git repository for enNCOMMAS. We can now push this up to the Dropbox repos by:

```
cd enNCOMMAS
git remote add dropbox file://$HOME/Dropbox/repos/enNCOMMAS.git
git push dropbox master
```

Inside the git directory, if you want `git pull` and `git push` to use the dropbox repo by default, you can type:

```
git config branch.master.remote dropbox
git config branch.master.merge master
```

Now you can use git just like SVN, e.g., say you have a new file you want to add:

```
git add new_file.f90
git commit new_file.f90
```

and the usual commit interface message comes up. For changes to an existing file,

```
git commit existing_file.f90
```

Now just because you have created a local commit – you have not yet "pushed" the changes up to the master. To do this now, simply:

```
git push
```

On another machine, you can simply clone from the dropbox account:

git clone -o dropbox file://$HOME/Dropbox/repos/enNCOMMAS.git

and if you already have the enNCOMMAS repos on your machine, you can pull the changes from the master:

git pull

In this way, you have the entire history of the enNCOMMAS on your machine and a master branch at Dropbox. One can then (at a finer grain level) control what is stored locally in a repo, and what is available to everyone – you have the entire history stored on you machine (like a laptop) and can branch/fork locally to test stuff, then merge back.

Some of this information was obtained from Bradley Wright's web page:

http://tumblr.intranation.com/post/766290743/using-dropbox-git-repository

Additional Techniques

Since we have local Git servers and you want to push code to a remote machine that may be outside the firewall – this is possible via the following procedure. Similar to the Dropbox procedure – you will create a bare local repository first, push your repository (or branches from it) to that remote repo, and then pull the code from the remote repository onto the remote machine. So on the remote machine, do the following commands

```
mkdir -p repos/enNCOMMAS.git
cd !$
git --bare init
```

You now have a bare repository. Now go back onto your local machine where the git repo sits, and type in:

git push ssh://remote_machine/…/…/repos/enNCOMMAS.git master

This pushes the code over to that machine. Now inside the enNCOMMAS.git directory, there is a repo (HEAD, config, etc.). This has all the changes you have on the local machine (which, BTW, does not have to match was is stored in the community dropbox). To get the code for the remote machine out of the repo,

```
cd repo
git pull
```

and wonderkind! In ~/repo/enNCOMMAS is your code. If then you change anything on the remote machine, commit it there and then you can PULL it to your local machine via:

git pull ssh://remote_machine/…/…/repos/enNCOMMAS

Note: you are pulling from the working copy, not the repo! You can still push changes from the local copy in the same way you did the initial push.

If you get a message when trying to push the repo from your local machine to the remote machine like:

error: refusing to update checked out branch: refs/heads/master

error: By default, updating the current branch in a non-bare repository

error: is denied, because it will make the index and work tree inconsistent

error: with what you pushed, and will require 'git reset --hard' to match

error: the work tree to HEAD.

error:

error: You can set 'receive.denyCurrentBranch' configuration variable to

error: 'ignore' or 'warn' in the remote repository to allow pushing into

error: its current branch; however, this is not recommended unless you

error: arranged to update its work tree to match what you pushed in some

error: other way.
error:

Then go to the remote machine and type in the enNCOMMAS.git repo,

git config receive.denyCurrentBranch ignore

and afterwards, you may not see your files, and type

git checkout –f

But as far as I can tell, you should not need to do this.

Handy Stuff

To commit all new changes (adding, deleting, etc), you can just type:

git commit –a

which is a big improvement over SVN