***Git Basics***

Scope:

Git can version-controlling scope is only below its parent directory (include all files and all subdirectory’s files (can have more and more subdirectories) by default. It shows like below:

dir

Dir1

Dir2

.git

F1

F2

Dir11

Dir121

Dir12

F121

F111

F22

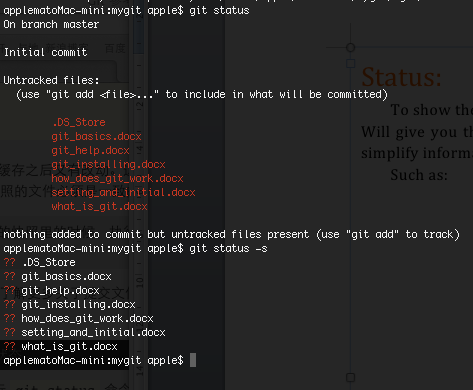
This shows that the git can only tracing the read-path (green file(s) or directory (directories)). It means all of content must below the “Dir1”.

Status:

To show the status between workspace and stage, using “git status”. “git status”

Will give you the full information about the status, and “git status -s” will get the simplify information.

Such as:



This will tell you under the “Untracked files” list is all files that you hadn’t been traced, and the ‘-s’ argument’s result shows in front of the file if have “??” it means that the file did not had been traced. And it tell you that to trace any files use “git add <file>” to include in what will be committed.

More about “git status” will be explained in “git add” or “git commit”, or other case (other command(s)).

Add

After you created a git-repo, if you want to trace the file(s), first of all you need to add the file(s) to the stage. You can do that like below:

//Add specify file

git add [path]file\_name

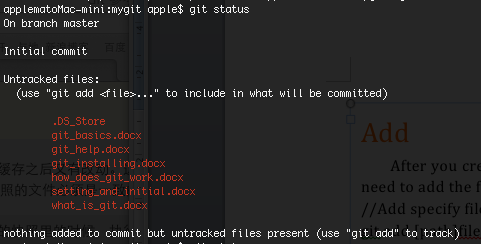
//Add all files included in the director

git add [path]dir

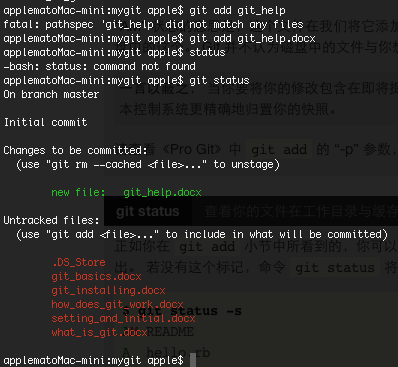
//Add all files in current director which file’s extends name is .c

git add \*.c

Before all of “add” you maybe need to check the status.



Now, It shows that no file had been traced, so add a file in the stage.

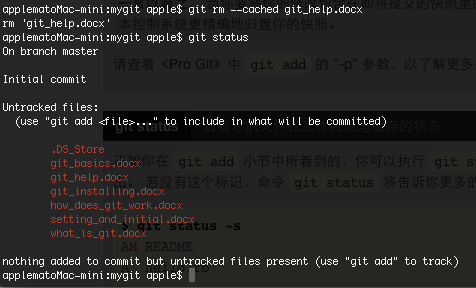


This shows that,

1 you must type the full name of the file.

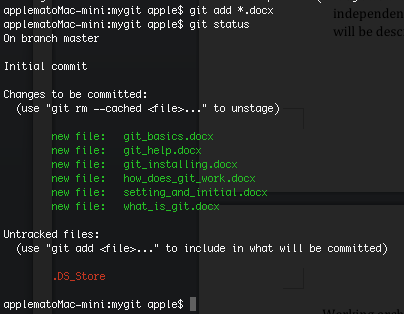
2 If any other error occurs, you would get the relation information with the command, other wise no extra information been showed.

3 This shows that the “git\_help.docx” file is new for tem-git-repo (stage), and already adds to stage (did not in actually git-repo), you can remove it from the stage. Why do not have a try?



Okay! The “git rm –cached file\_name” command do the work very well.

Now, use the wildcard character add all “.docx” files into stage.

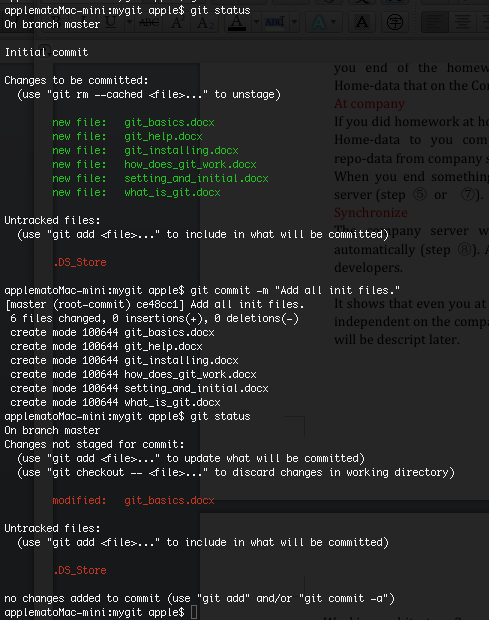


You can use the wildcard character remove all stage file also.

“Changes to be committed” tell you commit the changes, the “commit” had been explained below (see “Commit”).

Commit:

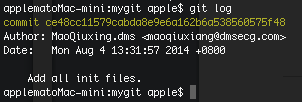
After you add all files you want to trace into the stage, if you want to actually save the stage content (all files) to the git repository, then use “git commit –m “Something commentate statements”, or “git commit” and then Git will open an editor to let you type the comments. This means that, you must use “git commit” command to save the stage to disk and must type the comment.



After “git commit” if no error occur, it will give you the message about the “commit”. Such as count of file, insertions rows, deletions rows, activity (here is create), and on which branch, and the unique ID, of course the comment.

After commit, use “status” can see the status between workspace and stage, because I write something in the git\_basics.docx file after add in to stage, so the status shows that git\_basics.docx had been modified, and any other files I did not edit, so nothing been showed.

The unique ID been consist of 40 characters, it only show some characters before of the full characters. If you use the “git log” command you will get the full characters, it shows like:



You can see, the given characters by commit is the sub-characters of log’s.