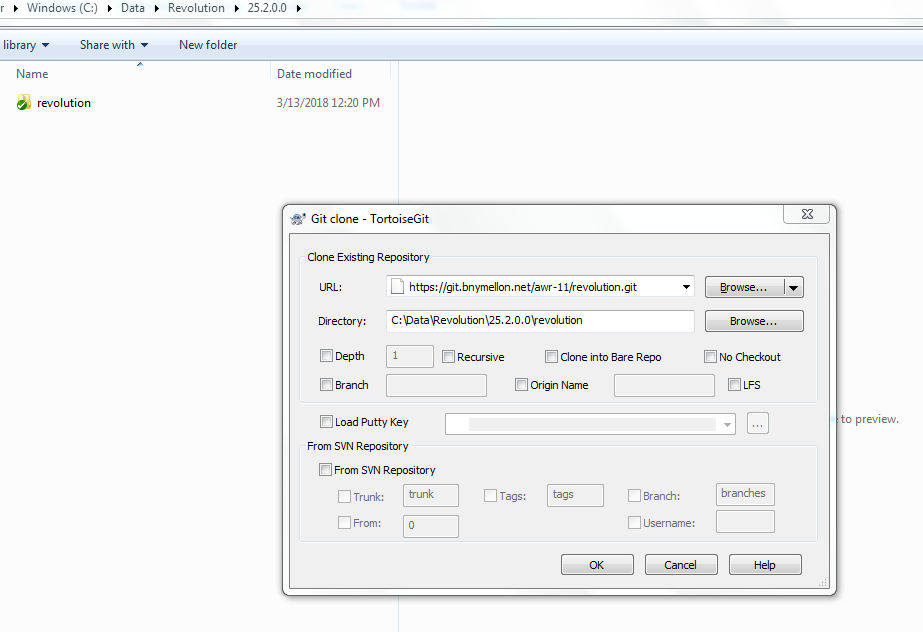
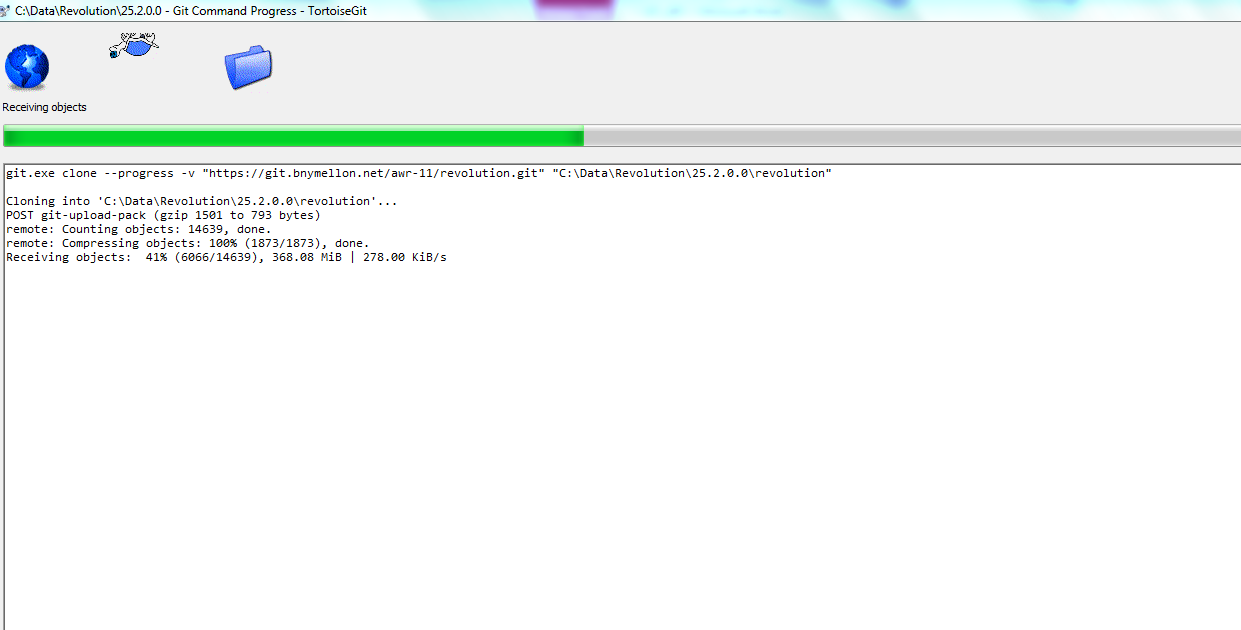
WHAT IS GIT?

**A Git repository is a virtual storage of your project. It allows you to save versions of your code, which can be accessed when needed.**

GIT CLONE:

* The **git clone** command copies an existing **Git** repository.
* This is sort of like SVN checkout.
* It has its own history, manages its own files, and is a completely isolated environment from the original repository.





BRANCH:

Branch is essentially an independent line of development.

You can take advantage of branch when working on :

1. new features
2. bug fixes

as it helps to isolate your work from that of other team members.

MASTER BRANCH:

Upon making the first commit in a repository, Git will automatically create a **master branch by default**. Subsequent commits will go under the master branch until you decide to create and switch over to another branch.

It is a common **practice to create a new branch for each task** (eg. bug fixing, new features etc.), which is a good practice because it allows others to easily identify

* what changes to expect,and
* also for backtracking purposes

to understand why a particular code change is implemented.

TO CREATE BRANCH:

$ git branch <branchname>

TO SWITCH BRANCHES:

Use the checkout command to switch branch.

$ git checkout <branch>

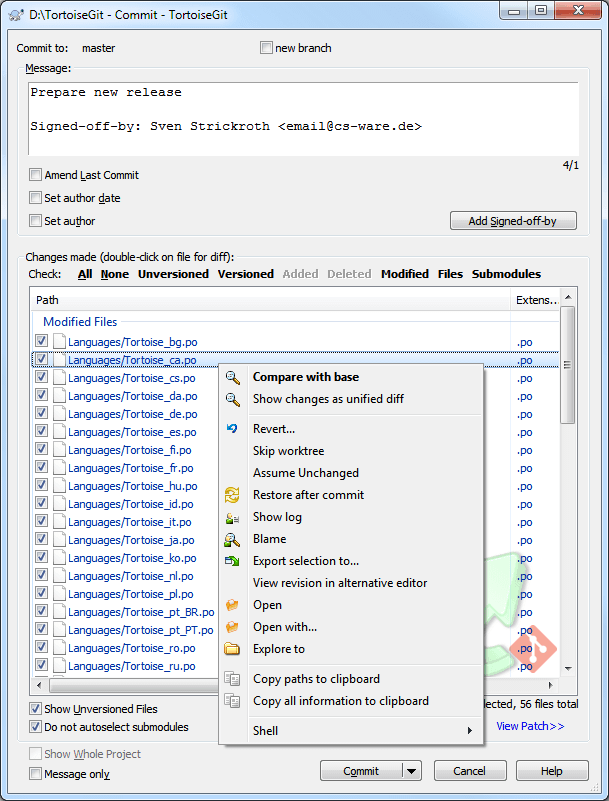
Switch to the branch "issue1" by doing the following.

$ **git checkout issue1**

Switched to branch 'issue1'

COMMIT and PUSH:

Basically **git commit** puts your changes into your local repo, while **git push** sends your changes to the remote location.

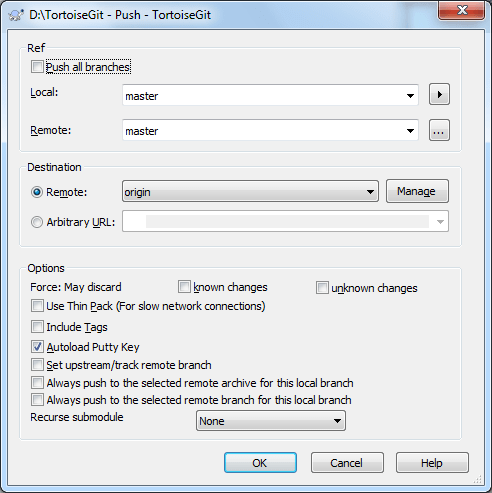


**PUSHING:**

**PUSH COMMAND:**

All of your commits are available to you until you push your local branch to the remote repository. That is, you can work on your own local branch at your own time and pace without affecting other members of the team.

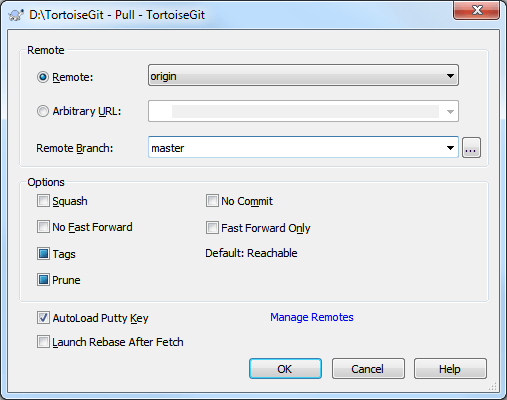
When you push your local branch to remote, Git will do a fast-forward merge to the destination repository.



**PULLING:**

**PULL COMMAND:**

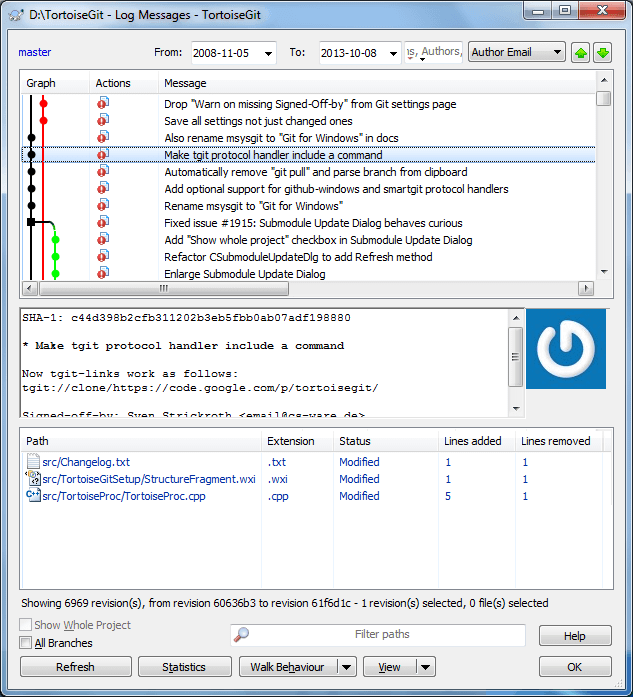
To **apply the latest changes** from a remote repository onto your local



**LOG DIALOG:**

Right click->tortoise GIT->Show Log

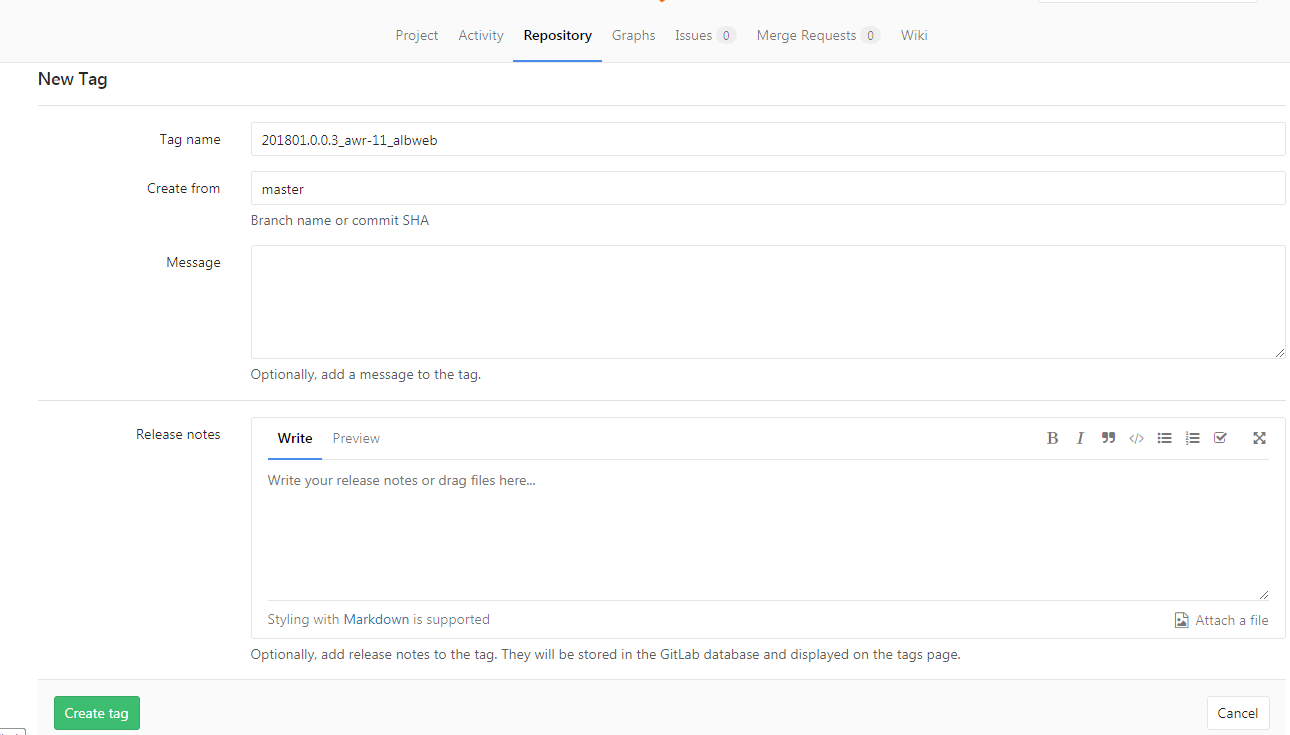
* able to find the changes made recently on that folder.



**TAG:**

A tag is used to **label and mark a specific point/commit** in the history. It is usually used to mark release points (eg. v1.0, etc.)

Although a tag may appear similar to branch, a tag however does not change. **It points directly to a specific commit in the history.**

****