Landis+Gyr
manage energy better

Working with git (Tortoise)

Release 1

Date: Today
Document Number: XXX
Filename: How to use Git.docx

Revision History

Issue	Date	Comments
0.1	19/03/2015	

Subject to change without notice

Landis+Gyr

www.landisgyr.com

Table of Contents

1 Introduction

A basic guide to using git with the tortoise git client tool.

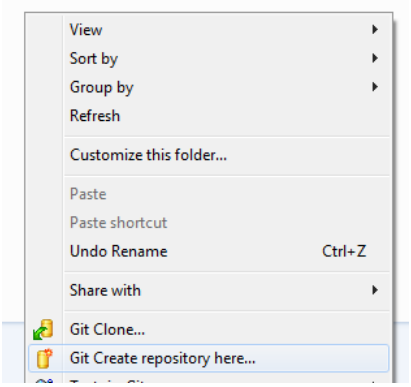
2 Installation

Download tortoise Git, install –

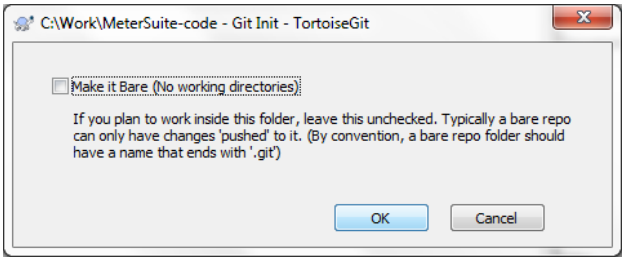
https://code.google.com/p/tortoisegit/wiki/Download

Create a folder to store the source code

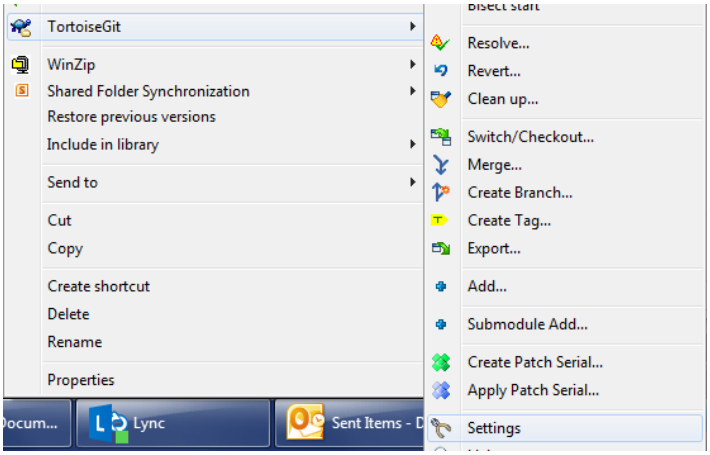
Rick click and select ‘Git Create repository here’



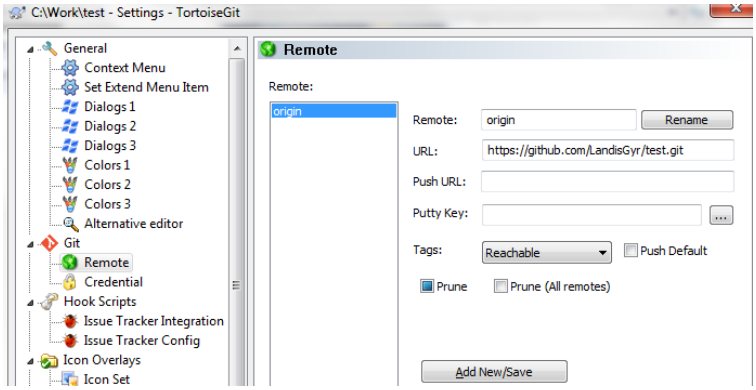
Select ‘ok’



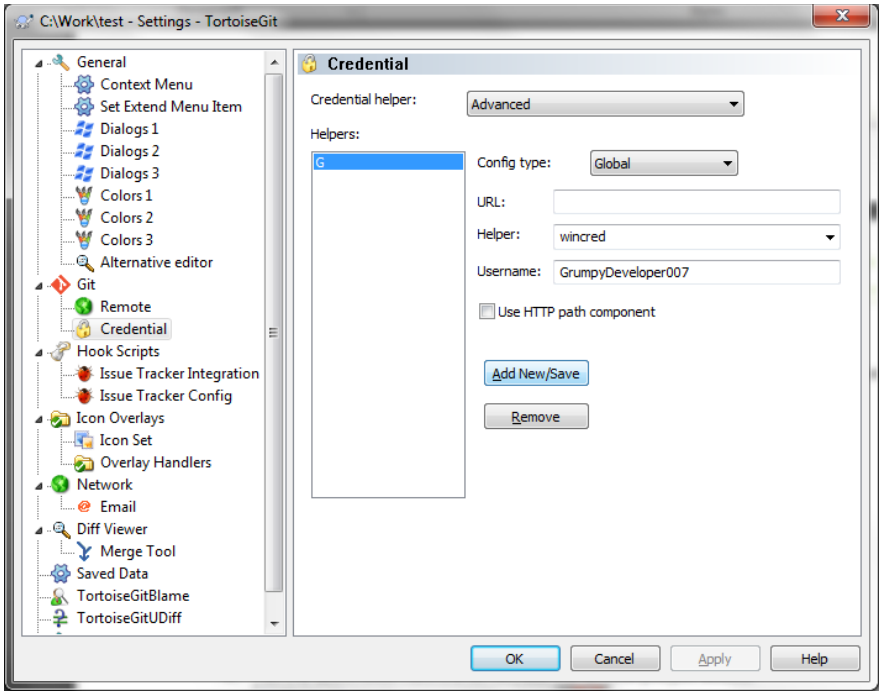
Configure a remote, click on setting for the folder you just created.



Add the address for the git repository in the remote selection, click on ‘add new/save’

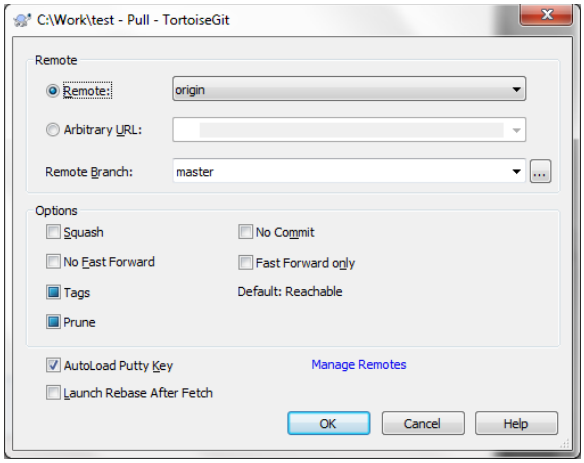


In the settings screen add a credential helper, this will make tortoise remember your user name.

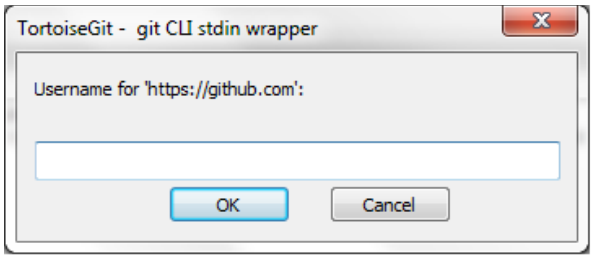


Right click on the folder, and then click on “git pull” to pull the latest code from the server. (Note: this menu item may sit above to tortoiseGit or stay in tortoiseGit submenu)

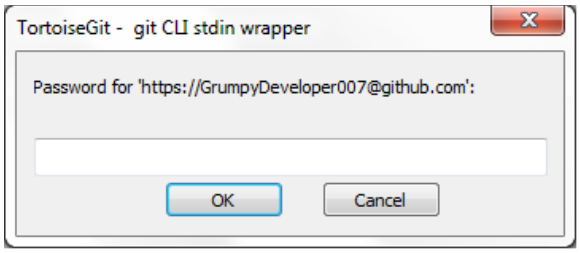
Use the remote you just added.



Enter user name



Enter password

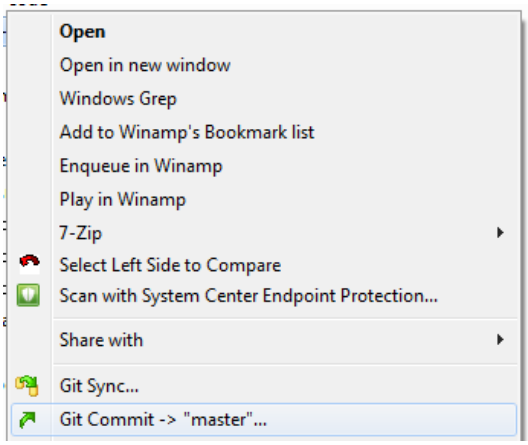


Done

2.1 Checking in your code

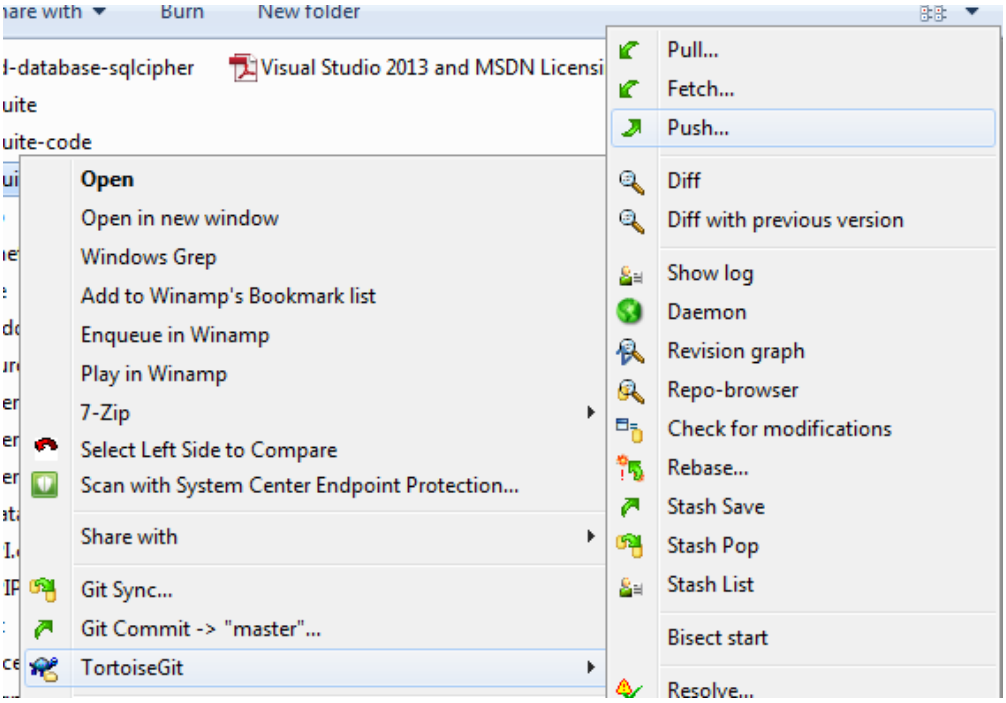
Once you are ready to ‘check in’ there are 2 steps required. Right click on the source code folder and select ‘git commit’

The first step only updates your local copy of the database.



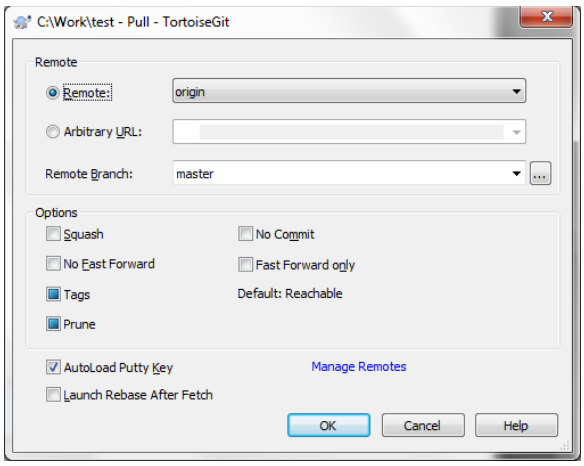
In the commit window, you can selected the files you want to commit and also put into comments.

Next do a ‘git push’, this will update the server.



2.2 Getting the latest code

To get the latest changes from the server, select ‘Pull…’

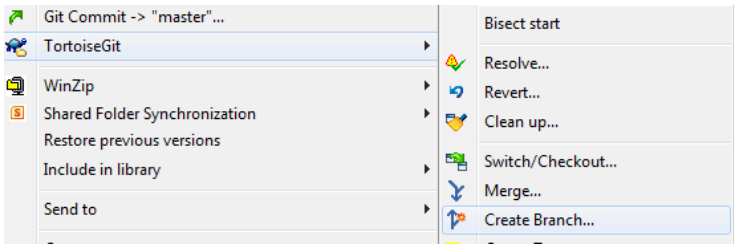


3 Feature branch

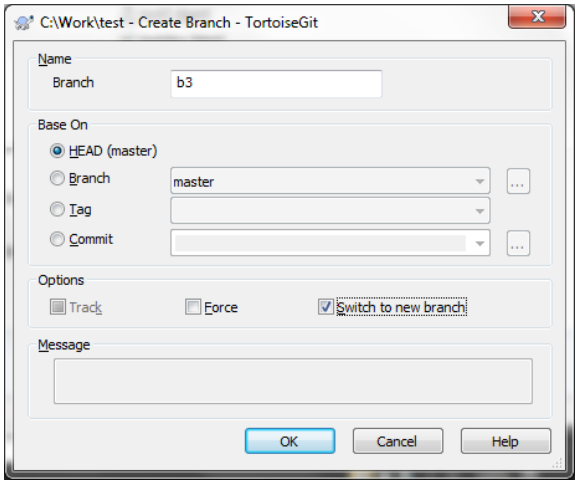
Feature branches allow developers to commit and push multiple change sets and push them without making changes visible to other uses.

3.1 Creating a new branch

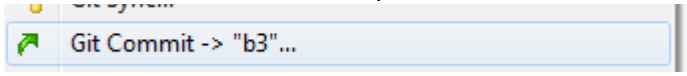
- Click on create branch



- Enter branch name, click on switch to branch

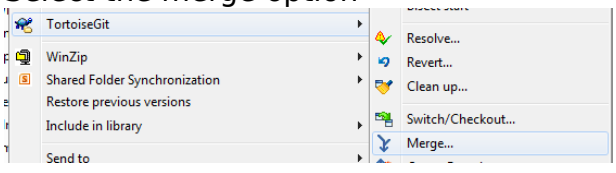


- Work as normal (commit/push)

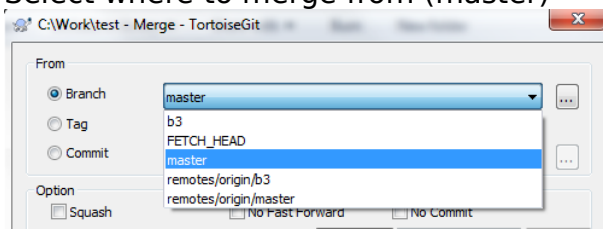


3.2 Merge master to the branch

Select the merge option



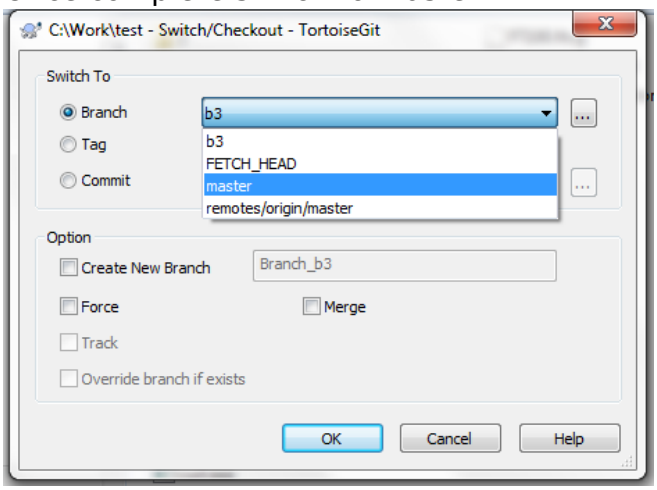
Select where to merge from (master)



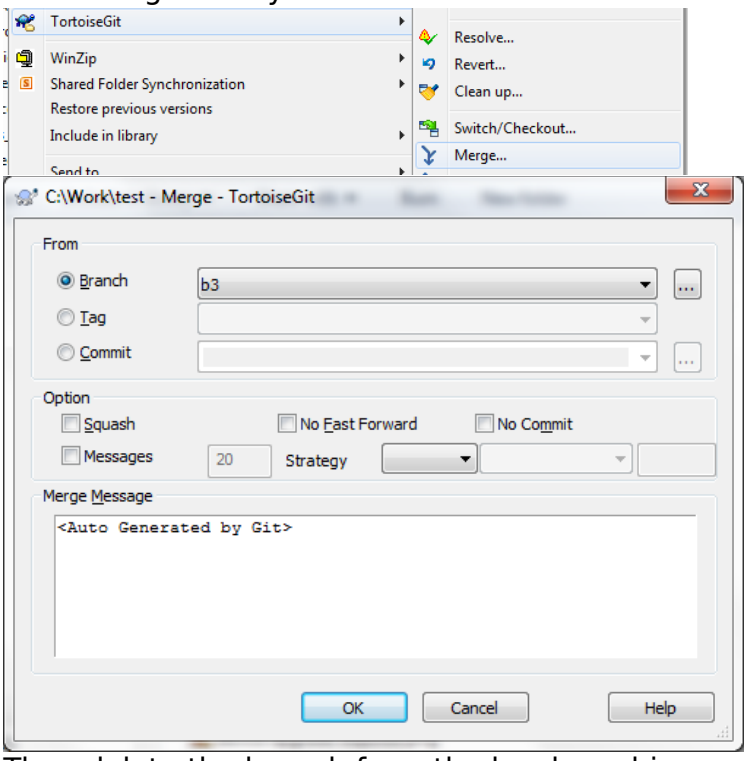
Done

3.3 Returning to master

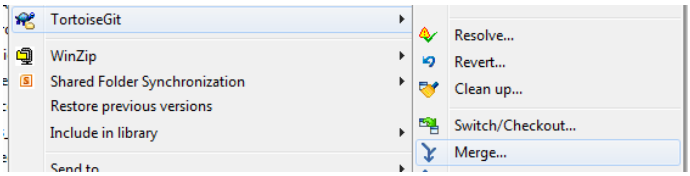
- Once complete switch to master



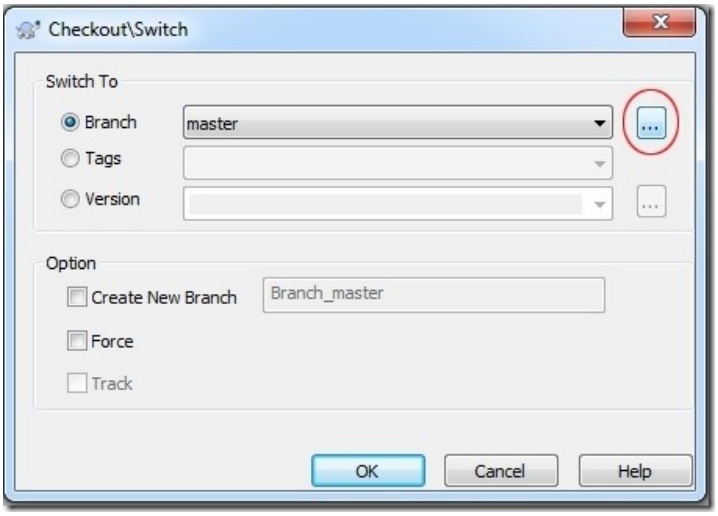
- Then merge from your branch



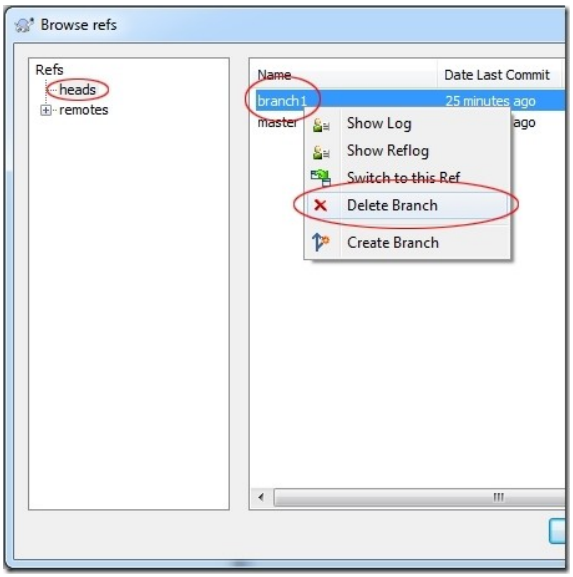
- Then delete the branch from the local machine and server



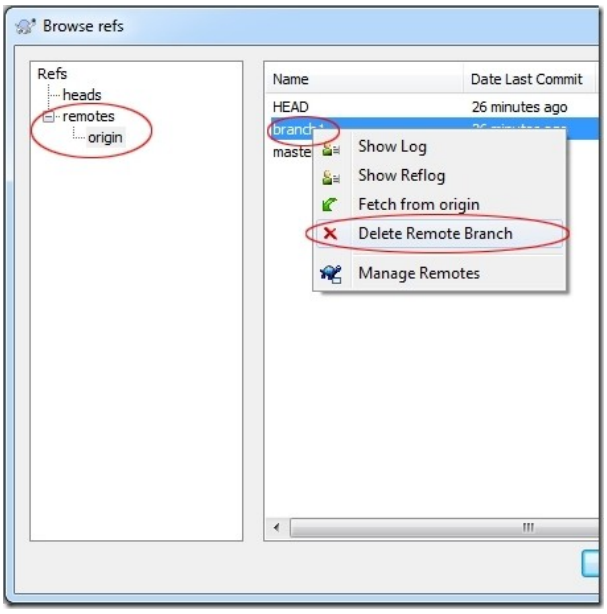
- ...remove the local branch by first opening up the Checkout/Switch dialog to get at the Browse refs dialog.



- In the Browse refs dialog we can right click on the local branch and choose to delete it.

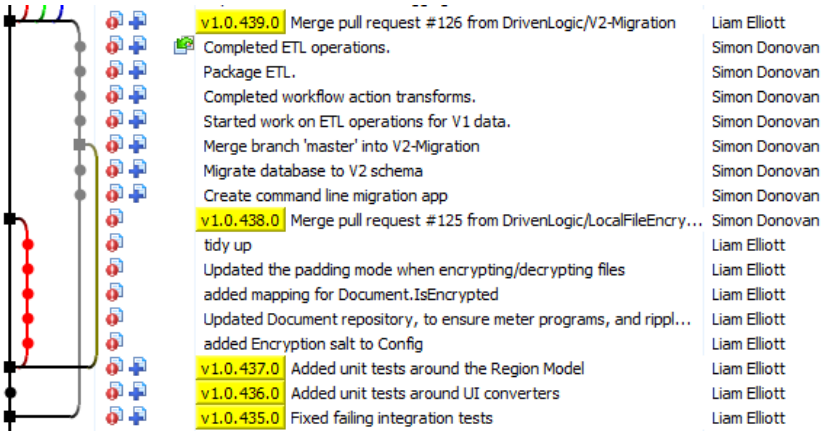


- To delete a remote branch we can do the same thing, but instead of right clicking on our local branch we expand the remotes tree in the left part of the dialog and then locate the remote branch.



3.4 Branch example

In the example below 2 branches were created, both users worked on their branch in insolation making multiple check-ins. After the work is completed the branch is merged in to master and deleted.



Without branches merge operations are more frequent and care must be taken to avoid causing problems in the others persons work.

4 Typical process for software development

1. “git pull” to get the latest code from git server.

2. creates a feature branch with a meaningful name indicating the dev work, for example “FT100 supports EM1210 full program”, if we have jira item, jira item number and title are recommended to be used as the branch name.

3. starts the dev work, and commits all your changes into the feature branch by commit and then push as many times as you want.

4. ~~you~~You can pull from master from remote server any time, and it will merge the master into your local feature branch, recompile to make sure the code is still working.

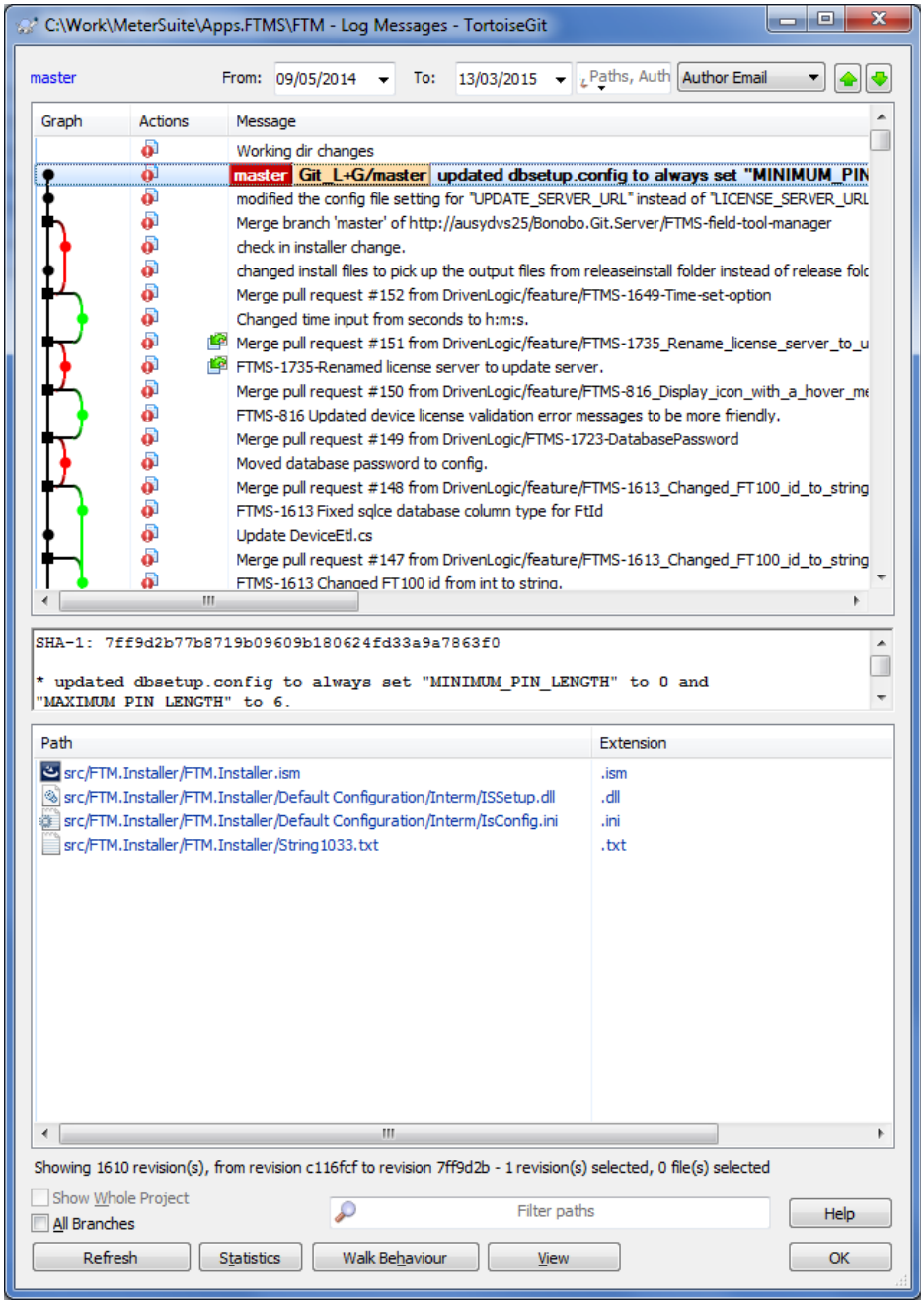
4. ~~when~~When the dev work is done, do a ~~last~~ pull from master ~~pull~~ to bring in all the master branch changes, recompile to make sure ~~its~~it’s all good and then commits all your changes into the feature branch.

5. You can trigger “pull request” **(in GitHub)** and this will send the request to other developers to review your code~~.~~ (not must do but encourage to do).

6. ~~after~~After reviewing, then go to merge your feature branch to master branch and delete your feature branch.

Note: this rule is not required for some minor bug fixings to the master branch. Bug fixings can happen on master branch unless it’s a complicated solution and requires some structure change.

TortoiseGit->show log is a very handy tool to use. It shows all the activities on each branch, and gives user very clear view.



5 Git Test project

A test repository has been setup (https://github.com/LandisGyr/test.git) this project can be used to test any of the above processes.