

Git/HgWorkbench Agile environment for software development

Author: David Huang

Date: 26, June, 2015

Introduction

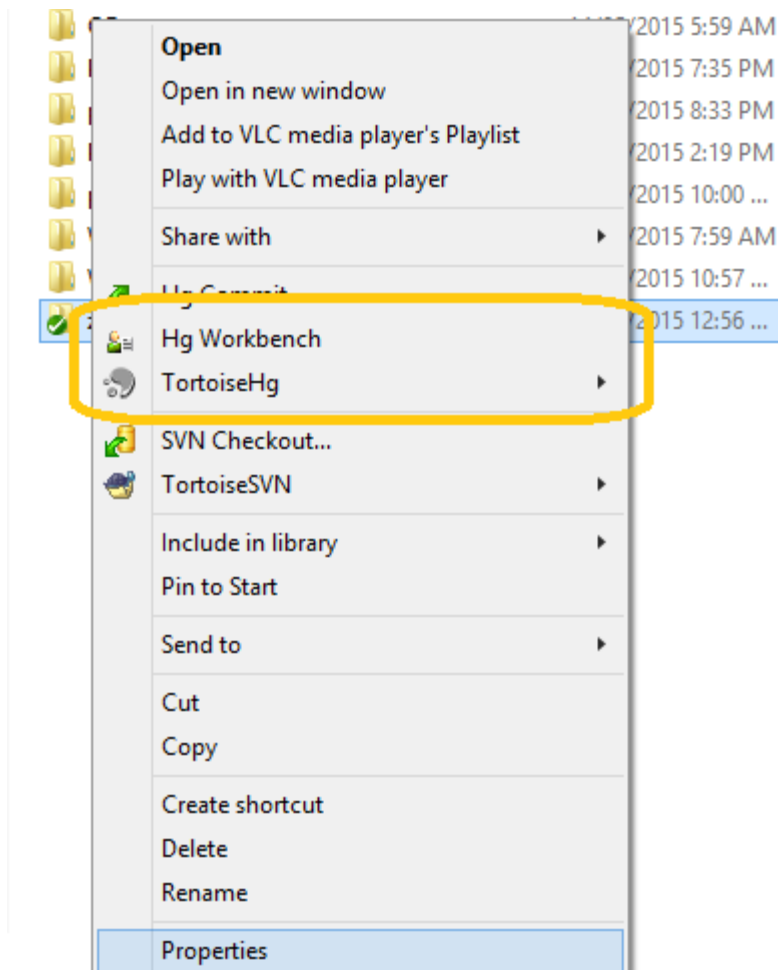
I once had a chance to work in git/Hgworkbench agile environment as a c# developer contractor. At beginning, I get confused the way of Hgworkbench management. Therefore, I download new version of Hgworkbench from <http://BitBucket.org> and finally figured out how it worked in my own laptop and remote repository in internet. I once installed a git subversion source control for github.com. It is not as flexible as this Hgworkbench. Now I loved to use this version control for my own IT projects. I then need to back up how this agile technology worked under my own IT environment.

Local Repository

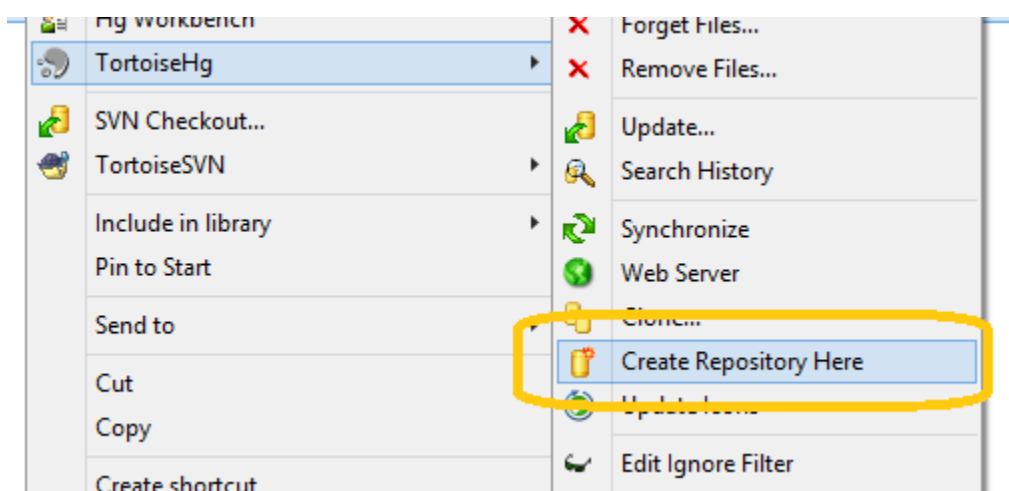
Every PC has a file directory in its local hard disk. Before we installed Git, those local directories are not controlled properly. You make changes on some files today; tomorrow you may forget those changes, so you have to spend a ton of time to recall your memory. This is not effective way for use to manage the files. Source control git and TFS software will help us to remember our work history and allow us to retrieve what we changed quickly. TFS now can be used online. However, I would like to install git for local source management.

How to get started

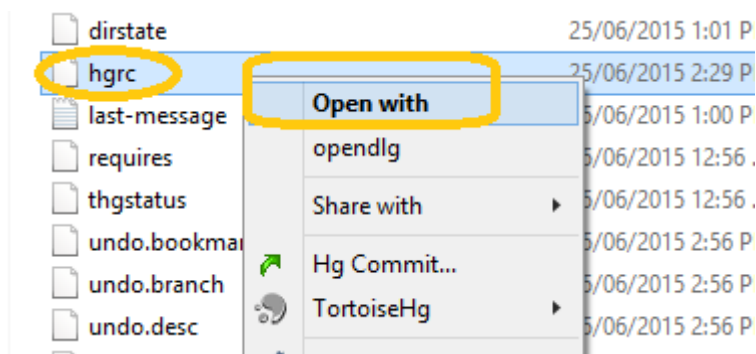
After we install git and Hgworkbench, we can see the changes in our file directory as one example shown below



At first, we want git to control this folder for us, so we need to create a repository in this folder, click TortoiseHg and select “create a repository here”, Git now creates a repository for this folder after adding some files inside this folder, this folder now is registered in Git and managed by Git. See picture below



Secondly, we open .hg folder inside this registered folder and update some settings for this folder so Git can use those settings to communicate with remote Git repository where we can upload the changes to and track the change logs. Basically we need to add some critical settings as below



```
[auth]
prefix = https://bitbucket.org/David/...
username = david@bitbucket.com
password = ...

[paths]
default = https://bitbucket.org/David/...

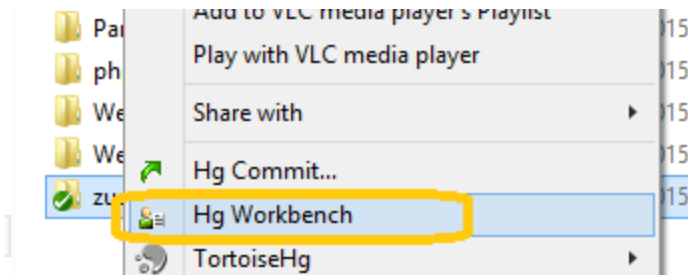
[web]
name = remoterepo
encoding = utf-8
description = localhost remote repo
style = paper
allow_archive = zip
allow_push = *
push_ssl = False

[email]
to = david@bitbucket.com
```

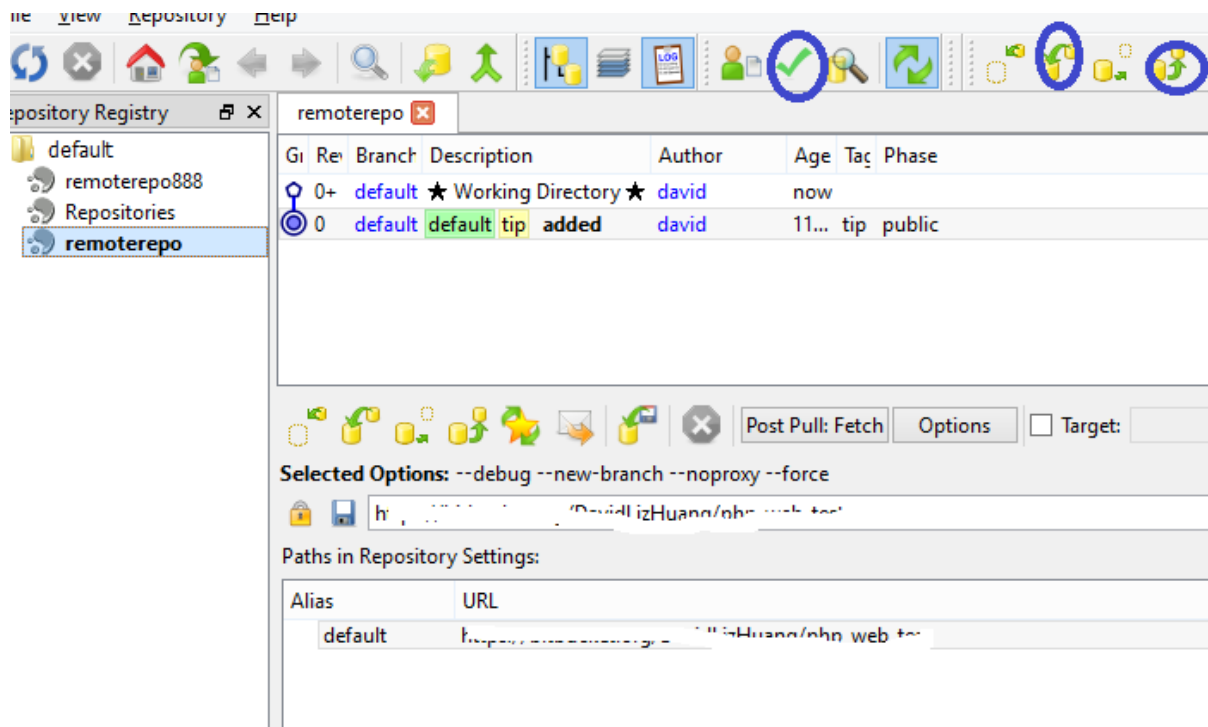
Now our local Git repository is ready to communicate with remote repository in internet.

Thirdly, now it is our time to register a new account in <http://www.bitbucket.org> web site, we then can create a repository under our own account. Copy this URL and add this URL to the hgrc file in our local file repository. This is it. It is our time to push our first version of software source codes to this Git repository in bitbucket.org web site.

Fourthly, Go to local repository, right click and select HgWorkBench see below



This can open the workbench for us to push codes to BitBucket.org.



Finally we can click commit button to commit the changes to local git repository and click push button to push our local repository to remote repository. Remote repository then can be accessed by your teams or customers after you invited them to join your repository. Now we can work together as a team for one project remotely. Team members can access to remote centre repo to download source codes and work locally to make the changes or publish the release version of software locally. It is really a nice technology we can easy to use.

Summary

Using this Git/HgWorkbench, we can do a lot of things such as view the changes, reverse the changes, and compare the changes, etc.

end