

// First Time Users integrating QT with GitHub

1. Install ssh-askpass (required by QT)
//note had to install ssh-askpass to get it to work with QtCreator
`$ sudo apt-get install ssh-askpass-gnome ssh-askpass`
2. create GitHub working directory (assumes you're somewhere under home dir. So
`$ cd ~/`
`$ mkdir GitHub_QT_Dev`
3. change dir into your working GitHub directory
`$cd GitHub_QT_Dev`
4. Initialize GitHub
`$git init`
5. Configure GitHub (include quotes along with "Your-Name-Here")
`$ git config --global user.name "Your-Name-Here"`
`$ git config --global user.email your.name@xxx.com`

//add editor (so we can add notes to each commit; req'd)
`$ git config --global core.editor gedit`

//to check configuration settings
`$ git config --list`
6. Clone Git Repository into Directory
`$ git clone https://github.com/Cotton-Engineering/Plastic-Contamination-Src`
7. Add all files into your local Git Repository (required)

// then cd into dir with all src files
`$ cd ./Plastic-Contamination-Srce`

//then add all files to local Repository (required)
`$ git add -A`
8. Open Project in QT. QT should now be fully operational with Git [assumes valid QT project was downloaded from GitHub Repository;
<https://github.com/Cotton-Engineering/Plastic-Contamination-Src>]
9. To use Git inside QT; goto menu Tools-Git; for more info. See this youtube video: <https://www.youtube.com/watch?v=q1PSV7Bs3rE>

Example: pull update down from within QtCreator using QT project: Plastic-Contamination-Src project that was previously setup, per above instructions:

1. goto menu Tools-Git-Remote_Repository-Pull (this will initiate a pull down from internet GitHub repository and update your local repository. It will prompt you as to what to do with your local copy; best options are "discard" or "stash and pop" [use 2nd if you've made changes you don't want to lose; that way you can reload it later and fork it should you wish to continue that development].

2. That's it; you've now updated the project with the latest version from the GitHub repository.

// ***** more optional cmd line notes below *****

For linking to github with ssh; (not needed for QT)

//generate ssh key (use email for github account)

\$ ssh-keygen -t rsa -b 4096 -C "yourName@xxx.com"

passphrase: ZxyAbcEfg!

//start up ssh-agent

\$ eval "\$(ssh-agent -s)"

//next add ssh private key to ssh-agent

// ssh-add ~/.ssh/id_rsa

//add ssh key to Github account

// copy ssh key to clipboard

\$ sudo apt-get install xclip

Downloads and installs xclip. If you don't have `apt-get`, you might need to use another installer (like `yum`)

\$ xclip -sel clip < ~/.ssh/id_rsa.pub

Copies the contents of the id_rsa.pub file to your clipboard

//not sure if we had to do that; we did do this

//from cmd line, in working QT directory, check remote config

\$ git remove -v origin

//this should show links to github site

//next do a push from cmd line (it'll prompt you for username and password)

\$ git push origin master

//check

\$ git status

//make a change with an editor (like QtCreator)

//check status; should see it knows which files were changed

//have to commit changes to add it to local repository before you can update internet GitHub repository
\$ git commit

//to view log's last 2 entries (-p shows differences between commits)
\$ git log -p -2

//working with remotes

//show which remotes we have defined
\$ git remote -v

//to push changes to remote (origin is name of one of the remotes; typically default name given //by git)
\$ git push origin master

//To tag a stable version; use
\$ git tag -a v1.0 -m "Stable version"

//now lets say you've continued development and ran into a dead-end; to recover the last //stable //version; just recall it via tag version id
\$ git checkout v1.0

//alternatively you can undo a commit by using revert
//lets say we have 4 committs and we want to undo the last
//one that has id = 514fbe7
//just run revert to remove it from head. Note: it's still there if you want it
\$ git revert 514fbe7