

Using Git Version Control

Edinburgh Napier University

February 23, 2017

Abstract

Version control(VC) is an industry standard practise. You will not find a software or technical company not using it in some form. There are many different implementations of VC, each with pro's and con's. We have adopted GIT, which is the standard for open source projects. This is one part of the system, you also need somewhere to host your VC code repository, for this we use Github.com or Bitbucket.com

1 Quick Start: Getting and updating module content

If you are on a module which is hoisting content via github, this section is for you.

1.1 Cloning the code for the first time

Step one, find the repository(repo) webpage.

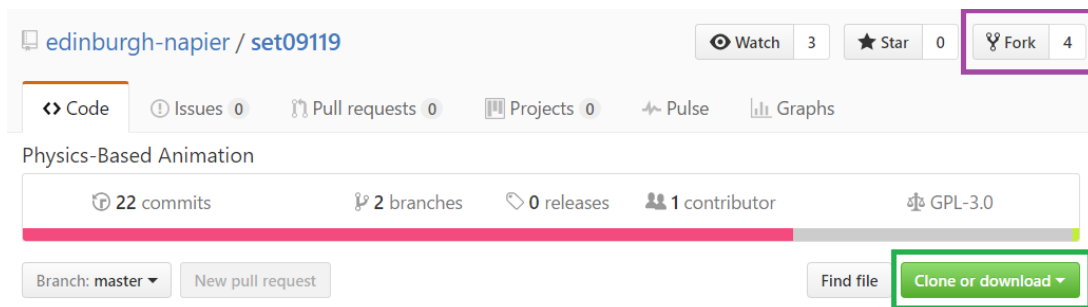


Figure 1: A github repo page for the content we wish to download

We now have three options:

1. **FORK** The best option – by pressing the fork button.
2. **CLONE** 2nd best, or only option if you want a private repo – Green clone button, Copy HTTPS link into sourcetree/gitbash or press open in desktop to clone with github desktop
3. **DOWNLOAD** Don't do this unless you need the file for some other reason, this will get you the files without any git information and you will have to create the repo manually. – Press Green clone button, then Download Zip

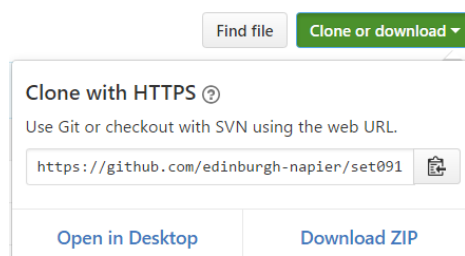


Figure 2: Clone or download options

Forking process Forking makes a copy of the repo within your own github account, this is equivalent to cloning the repo to your pc, creating the repo on github and then pushing it back up. It has the benefits that Github knows this is a fork and will update stats and metrics accordingly. If you ever need to do a pull request, this has to be done via a fork. You can now CLONE your fork down to your pc, work on it and push whenever. The one side effect of this is that Github does not allow for private forks, so if you wish to keep your work private, you must use manual clone method.

Cloning process Using your Git client of choice, clone the repo using the https link provided by github. Then create a new repo on Github.com, add this as a remote to your local repo and push to it. This is very similar to creating a repo from scratch, but you've already got some content.

2 Updating content

If the original module repo has been updated and you want to grab these changes, the process is:

2.0.1 Add the original Repo as a remote

Listing 1: gitbash ADD REMOTE command

```
1 git remote add updates https://github.com/edinburgh-napier/repo.git
2
```

Or using Sourcetree:

Repository > Repository Settings > Remotes Tab > Add button

Remote name: 'Updates'

Url / Path: 'https://github.com/edinburgh-napier/repo.git'

2.0.2 Pull the updates

Listing 2: gitbash PULL command

```
1 git pull updates master
2
```

Or using Sourcetree:

PULL > pull from remote: updates > press OK

2.0.3 Deal with any merge conflicts

Either manually edit the problem files to resolve, or use Sourcetree's 'Resolve>Keep theirs' feature if you only want to completely overwrite your local files with the new content (and possibly loose work).

2.0.4 Commit and push to your repo

And you are done!

3 Git workflow

Coming soon