{{lastupdated_at}} by {{lastupdated_by}}

GitHub repository configuration

To prepare for GammaLib development using GutHub you have to perform the following steps (you need to do this only once):

1. Create a GitHub account
2. Create a GammaLib fork
3. Clone your fork to your computer
4. Connect to the GitHub GammaLib repository
5. Deleting your master branch (optional)

Create your GitHub account

If you don't have a GitHub account, go to the GitHub page, and make one.

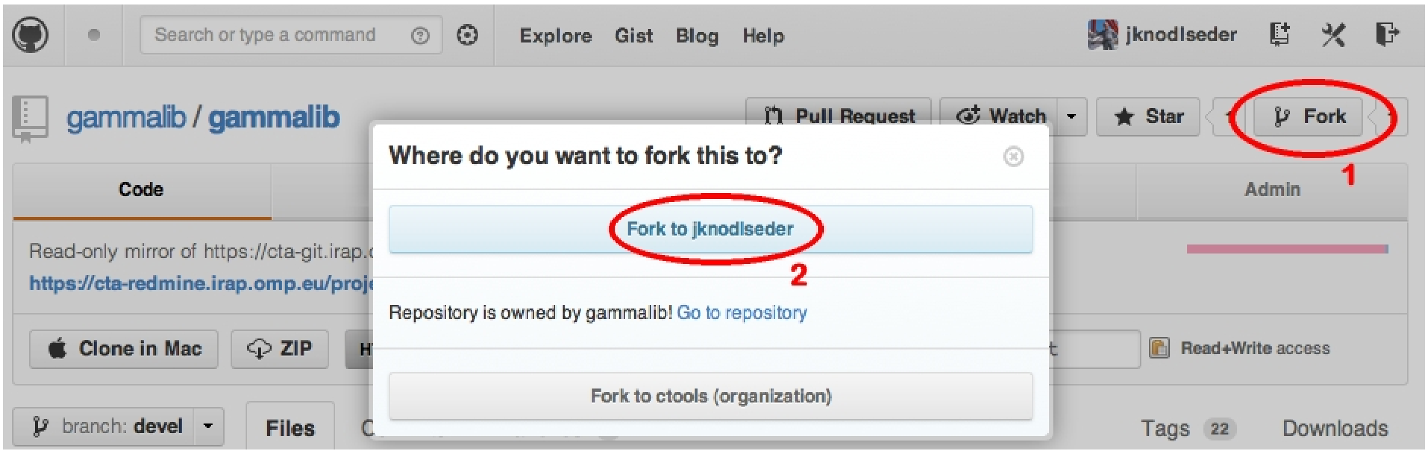
You then need to configure your account to allow write access - see the Generating SSH keys help on GitHub Help.

Create a GammaLib fork

As next step you need to create a fork of gammalib on GitHub. The instructions here are very similar to the instructions at http://help.github.com/fork-a-repo/ - please see that page for more details. We're repeating some of it here just to give the specifics for the GammaLib project, and to suggest some default names.

The following example shows how to fork the GammaLib repository:

1. Log into your GitHub account.
2. Goto to https://github.com/gammalib/gammalib
3. Click on the fork button
4. Select your username (in this example jknodlseder)



Where do you want to fork this to?

Repository is owned by gammalib! Go to repository

Read-only mirror of https://cta-git.irap.c

https://cta-redmine.irap.omp.eu/proj

After a short pause, you should find yourself at the home page for your own forked copy of GammaLib (in this example https://github.com/jknodlseder/gammalib).

Clone your fork to your computer

As next step, you have to clone your fork to your computer. Use the command

```
$ git clone git@github.com:user/gammalib.git
  remote: Counting objects: 22147, done.
  remote: Compressing objects: 100% (4911/4911), done.
  remote: Total 22147 (delta 17346), reused 21980 (delta 17179)
  Receiving objects: 100% (22147/22147), 80.25 MiB | 42 KiB/s, done.
  Resolving deltas: 100% (17346/17346), done.
```

to clone the GammaLib repository from GitHub. Here, user is your GitHub user name.

Connect to the GitHub GammaLib repository

Now connect your clone to the GitHub GammaLib repository so that you can fetch upstream modifications from the repo.

```
$ cd gammalib
$ git remote add upstream git://github.com/gammalib/gammalib.git
```

upstream here is just the arbitrary name we're using to refer to the main GammaLib repository.

Note that we've used git:// for the URL rather than git@. The git:// URL is read only. This means we that we can't accidentally (or deliberately) write to the upstream repo, and we are only going to use it to merge into our own code.

You may verify that the connection has been established with

```
$ git remote -v
  origin    git@github.com:jknodlseder/gammalib.git (fetch)
  origin    git@github.com:jknodlseder/gammalib.git (push)
  upstream  git://github.com/gammalib/gammalib.git (fetch)
  upstream  git://github.com/gammalib/gammalib.git (push)
```

Your fork is now set up correctly, and you are ready to hack away.

Deleting your master branch

It may sound strange, but deleting your own master branch can help reduce confusion about which branch you are on. See deleting master on github for details.

To delete the master branch, type

```
$ git checkout devel
  Already on 'devel'
$ git branch -D master
  error: branch 'master' not found.
$ git push origin :master
  To git@github.com:jknodlseder/gammalib.git
   - [deleted]         master
```

Don't worry if you get the message error: branch 'master' not found., this just signals that you never checked out the master branch.

You may do the same thing with the release and integration branches.

Files

fork-gammalib.jpg	138 KB	12/09/2012	Knödlseder Jürgen