# NEF on GitHup

The NEF is a freely-available public standard agreed on by several NMR groups around the world. Although its long-term stability is essential, it should be easy to raise issues, discuss those issues, propose and then integrate solutions into the standard in a transparent way. Many open-source software projects face the same set of requirements, and we have tested and implemented one of the most common (and free) solutions: Git and GitHub. Git, like its counterparts cvs and svn, allows for the controlled versioning of files. In contrast to cvs and svn, Git is more suitable for xxx as it yyy

## Read-only access

If you would like read-only access to the NEF, you can browse the repository at: <https://github.com/NMRExchangeFormat/NEF>. In addition, GitHub provides a link to download a complete copy of the repository to you local machine. Issues can also be raised by clicking on the ‘issues’ link at the above url.

## Short instructions for contributing to the NEF

The instructions are for people familiar with Git and GitHub:

1. Log in to GitHub and fork the repository at <https://github.com/NMRExchangeFormat/NEF>.
2. Clone the repository from your account to your local machine.
3. Choose which branch (or create your own,) you’d like to contribute to.
4. Push changes back to your GitHub account.
5. Create a pull request detailing your changes to the community.

## Longer instructions for contributing to the NEF

These instructions assume general knowledge of the command line. We strongly recommend to do your first setup following these instructions using the command line and reverting to a gui (if so desired) afterwards.

First, you need to setup your GitHub account.

1. Sign-up for a free GitHub (<https://github.com>) account. You can sign-up as a lab, but we encourage each developer to have their own GitHub account. The free account will suffice and there is no need to enter credit card details. This defines your “*My UserName*” and “*MyUsername@email.com”* used in step 3. In the settings, you can update your profile if you wish to do so.
2. In order to contribute to the NEF, you need to install Git (via your favorite package manager (or <http://git-scm.com/> This link also can direct you to gui-clients for the Linux, Mac and Windows platforms).
3. Setup git by issuing the following commands in the terminal:

git config --global user.name “*My Username*”

git config --global user.email “*MyUsername@email.com”*

Substituting an appropriate username and email address as defined in step 1.

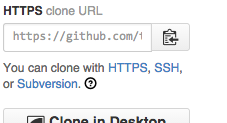
1. Once you’re signed-in to GitHub, navigate to <https://github.com/NMRExchangeFormat/NEF> and fork the project by clicking on the ‘Fork’ link near the top right-hand corner of the page:



This will make a complete copy of the official NEF repository in your GitHub account.

Now you need to make a working copy of the NEF repository on your local computer.

1. Make a directory on your local machine to store the NEF repository. The repo will be cloned into this directory from your GitHub account. So if you’re in the directory /Users/username/documents, the repo will get cloned into /Users/username/documents/NEF.
2. Go to **your** NEF repo on GitHub and click on the copy button to copy the HTTPS clone address:



1. In a terminal, change to the relevant directory and issue the command:

git clone *myURL*

replacing *myURL* with the copied URL from step 4.

You should see Git download your copy of the NEF repo to the current directory.

Change to the NEF directory in the terminal:

cd NEF

By convention, your GitHub repo is called ‘origin’, and the repo you originally forked from (NMRExchangeFormat/NEF,) is called ‘upstream’. We’ll stick to those conventions here.

1. In the terminal, define the upstream repo by issuing the command:

git remote add upstream https://github.com/NMRExchangeFormat/NEF.git

Git allows multiple light-weight branches to exist within one repository. Currently, we only have one branch called ‘master’, although this may change in the future. For now, all operations will be carried out in the master branch. Before moving any further, we’ll synchronize our local master branch with the NMRExchangeFormat/NEF (upstream) master branch, then push any changes to our GitHub account. Note: there will likely be no changes, don’t be surprised if it claims there’s nothing to do.

1. Retrieve all the changes in the upstream repo by issuing the command:

git fetch upstream

1. Make sure you’re on the master branch of you local repo by checking out master:

git checkout master

1. Now merge the upstream master branch into the current branch:

git merge upstream/master

1. And finally push the changes to the master branch to your GitHub account:

git push origin master

Now that your origin repo and local working repo are setup and synchronized with upstream, you’re ready to make changes. Please look at any one of the many Git tutorials online and/or read the manual (<http://git-scm.com/documentation>) to see what more can be done. For now, you’re just going to create a branch, edit a file in that branch and propagate those changes to you online repo. Finally, you’ll attempt to get your change into the upstream repo by issuing a pull request (which, for obvious reasons, will be refused.)

1. Create a branch called ‘myname’ and checkout the new branch:

git branch myname

git checkout myname

1. Using your favorite editor, alter the file ‘README.md’ by adding you name (or something else innocuous,) to the end of the file.
2. Add the file to the list of staged files in Git:

git add README.md

Note – you can also do ‘git add --all’ or ‘git add -A’ to add all files and folders recursively.

1. Commit your changes to your local Git repo:

git commit –m ‘*commit message*’

Note: Git requires commit messages. If you don’t supply one, you’ll be dropped into vi to provide one. If you get stuck in vi, type esc:q! and try again.

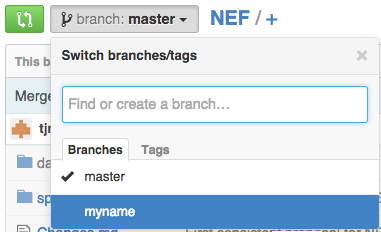
1. Push changes from the myname branch to GitHub:

git push origin myname

Note: any changes you push to origin (or anywhere else on GitHub) will be immediately public.

The last thing to do is to try to get your change integrated into the official NMRExchangeFormat NEF repository. (Note: this is only one of several ways to create a pull request.)

1. Go to **your** NEF repository on GitHub and change from your master branch to your myname branch:

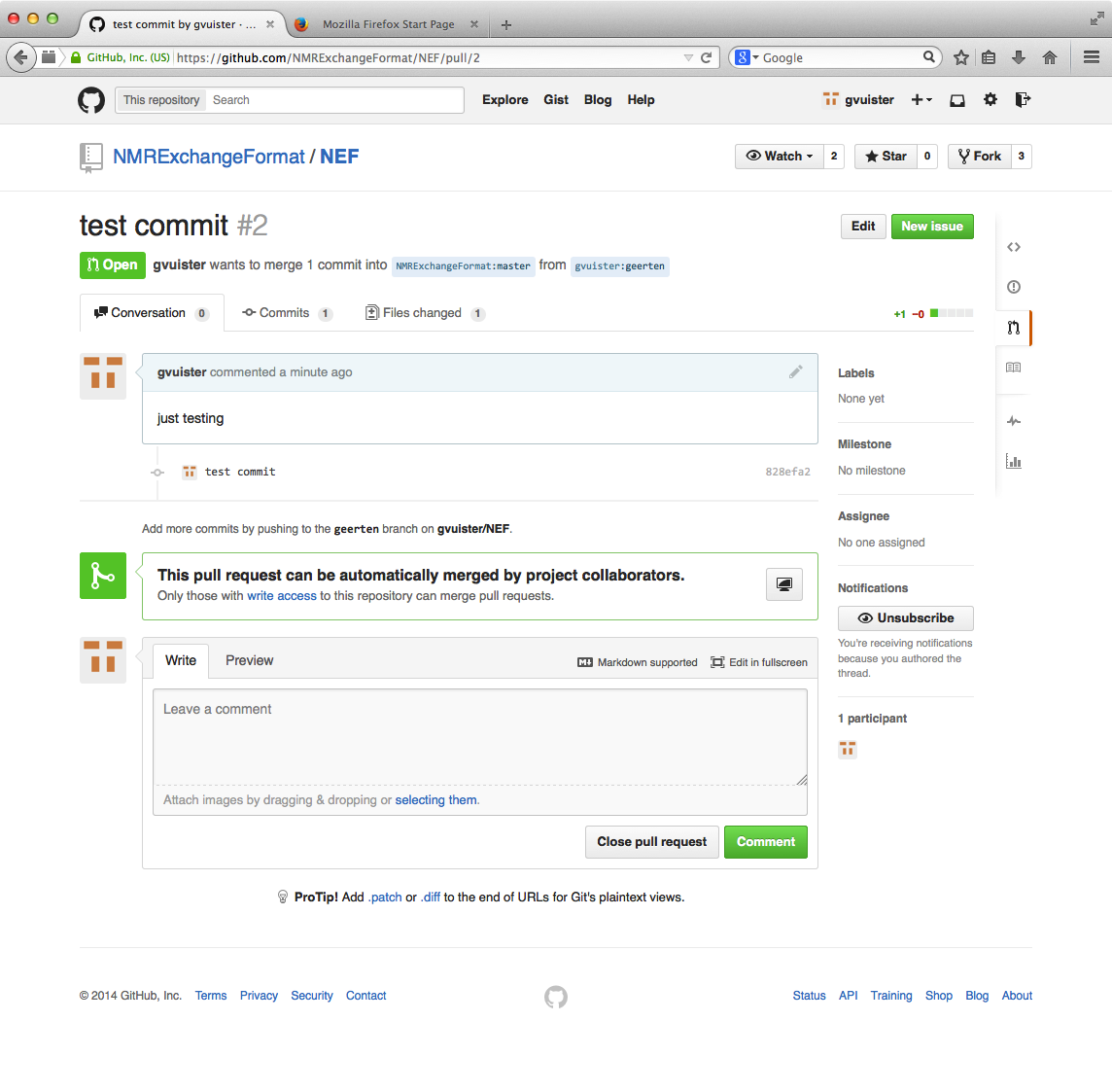


1. Click on the green ‘create pull request’ button on the upper left-hand side:



1. Examine the diff to see what changes you are about to propose. If you’re happy, click on the green ‘create pull request’ button:

Macintosh HD:Users:tjr22:Desktop:Screen Shot 2014-12-18 at 14.22.10.png



1. Enter a brief title for your pull request, and write a description of what the proposed changes are to start the discussion, and finally click on the green ‘Create pull request’ button.
2. The NMRExchangeFormat secretariat will put forward your changes for discussion.

Note: You can also generate pull requests to anyone else’s repo in exactly the same way. First, add their repo as a remote as in step 7, then edit the target repo listed at the top of the page that results from step 18. That way you can collaborate on changes without involving the official NEF repo until they are ready to be submitted.