

# OMFIT 1st Commit Tutorial

---

## **DISCLAIMER:**

This is an editable public document! When you make a change to it, it will be visible immediately from everyone! Feel free to edit it and help other OMFIT users!

Since this is an evolving document, there may be some small inconsistencies as different figures have been taken by different people with different versions of OMFIT for different analyses.

---

This tutorial assumes that you have run OMFIT before, and have made a change in a project or module that you now want to contribute back to others (YAY!). (If you want to make a change to something besides a module, you can skip the export step.)

[Overview of Github Workflow](#)

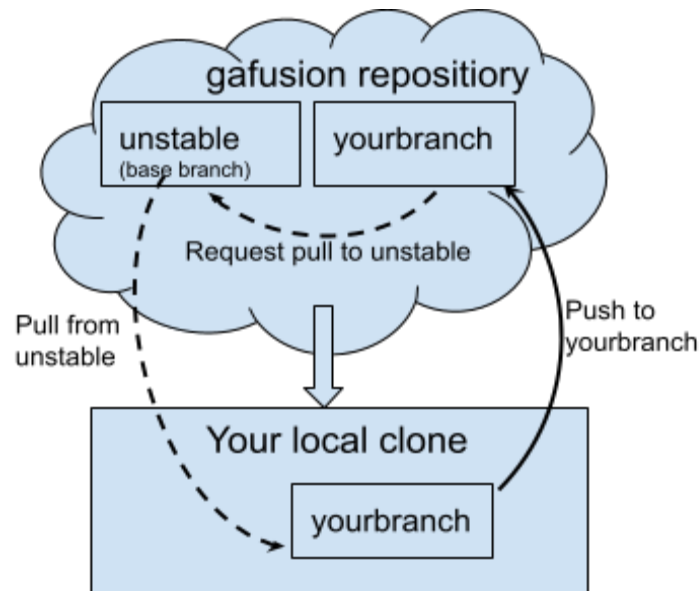
[Get your own copy of the OMFIT-source repository](#)

[Export your changes from your OMFIT session](#)

[Commit your changes](#)

[Appendix A: managing your own fork of the OMFIT-source repository](#)

# Overview of Github Workflow



In the simplest approach (explained in excruciating detail below), you could follow this flow:

1. On your local clone, update yourbranch to the latest unstable
  - a. `git checkout yourbranch`
  - b. `git pull -r origin unstable` (note that `git pull` is a combination command, alternatively do `git fetch gafusion` followed by `git rebase origin/unstable`)
2. Make changes, export the module, commit
3. `git push -u origin yourbranch` (the `-u` sets the upstream branch)  
(if upstream repo doesn't exist, then first use `git remote add` command)
4. Make pull request to gafusion unstable

To get help with commands, use `git <command> -help`, e.g. `git push -help`

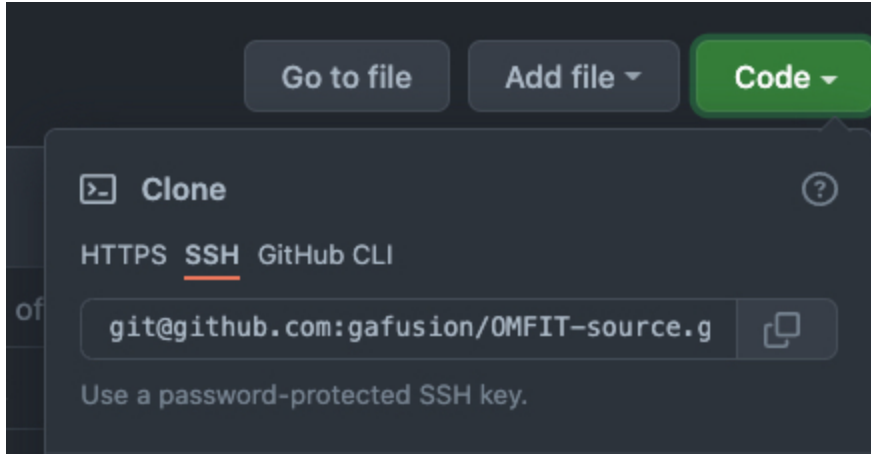
Also, see <http://gafusion.github.io/OMFIT-source/development.html>

`git branch -v` (to see your local branches and descriptions)

`git remote -v` (to see your remote repos)

# Get your own copy of the OMFIT-source repository

1. Generate an ssh key - `ssh-keygen -b 4096 -m PEM -t rsa`
  - a. On iris, ssh-keygen cannot generate the key using the preferred ed25519 algorithm described here: [Generating a new SSH key and adding it to the ssh-agent - GitHub Docs](#) . The legacy option they show also didn't work. Instead, this option worked:
    - i. `ssh-keygen -t ecdsa -b 521`
2. Register for GitHub.com
  - a. Set up your ssh key on GitHub.com (documentation at <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account> ; direct link to settings at <https://github.com/settings/keys> )
3. Register for GitLab.com
  - a. Set up your ssh key on GitLab.com (documentation at <https://docs.gitlab.com/ee/user/ssh.html>; direct link to settings at [https://gitlab.com/-/user\\_settings/ssh\\_keys](https://gitlab.com/-/user_settings/ssh_keys) )
4. Fill out <https://form.omfit.io/> including your GitHub (not GitLab) username
5. Verify you can access <https://github.com/gafusion/OMFIT-source/>
6. Copy the url for Code->Clone (use SSH like shown below).



7. On the machine where you are running OMFIT, in the directory where you want a new OMFIT-source directory containing the OMFIT git repository, issue git clone the following (**if you want to manage your own fork, skip to appendix A at the bottom of this document after following these steps**)

```
git clone --recursive -b unstable git@github.com:gafusion/OMFIT-source.git
cd OMFIT-source
git checkout -b name_of_new_branch
git config --global pull.rebase true
```

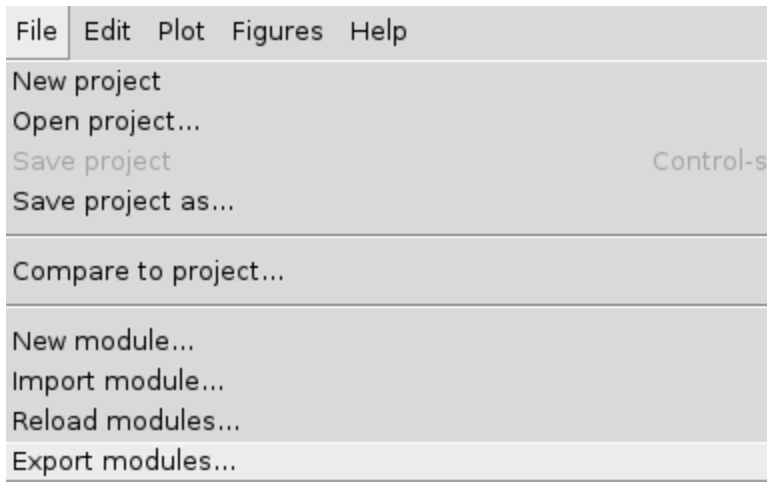
```
git config --global user.name "YOUR NAME"
git config --global user.email YOUREMAIL@EMAIL.COM
git config --global color.ui auto
```

## Export modules changes from your OMFIT session

1. Start your own copy of OMFIT in the git-updated directory you just pulled from GitHub and modify what you desire
  - a. If you are running on your laptop, run
 

```
`python OMFIT-source/omfit/omfit.py`
```
  - b. If you are running on a server that has a public OMFIT installation, you can start your own copy of OMFIT with
 

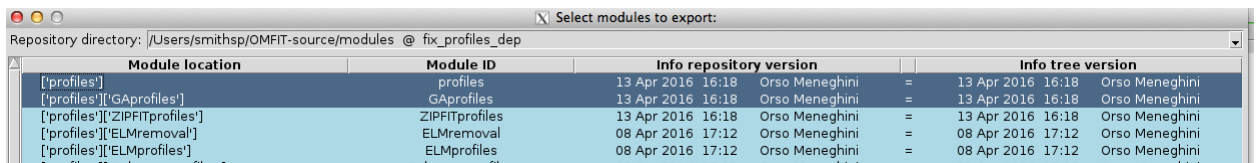
```
`cd /your/folder/OMFIT-source; omfit --cwd`
```
2. Go to File->Export modules



3. Make sure the repository directory is pointing to your repository:

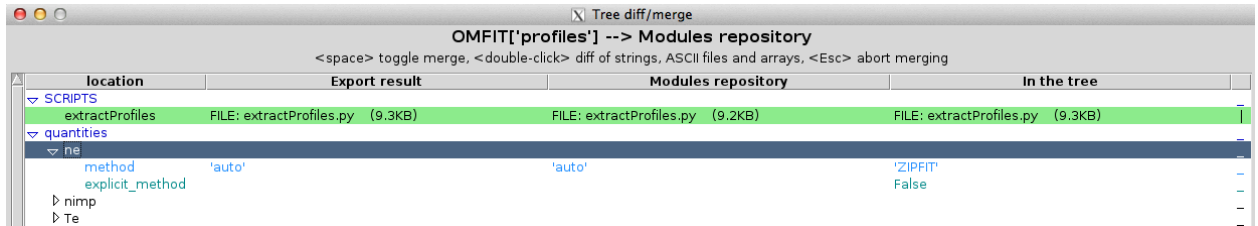
Repository directory: `/Users/smithsp/OMFIT-source/modules @ fix_profiles_dep`

4. Select the module you want to export. You can select multiple modules using <control-click>

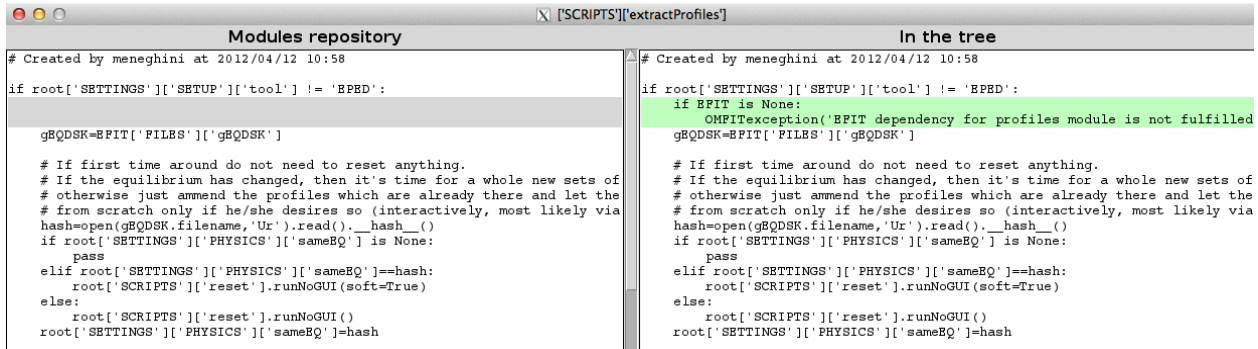


Then choose Export module(s) at the bottom

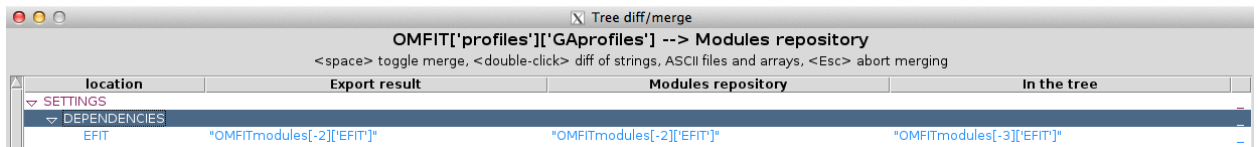
5. You are then presented with a tree view of the diff of your project and the repository. For each change you want to export, choose that line, then press spacebar



Note the line on the right will turn vertical and the background highlighting will be green for those items that will be exported. For some string and file objects, you can double click on the item to bring up a visual difference of the changes



Unless specifically directed, do not change dependencies such as shown here



6. Press “Merge Highlighted Entries” (press <escape> or close the window to dismiss)

## Commit your changes

7. Go back to the command line. Give the command ``git status``
8. If there are any untracked files, use ``git add`` to add them
9. Set the EDITOR environment variable to your favorite (blocking) editor. I use nedit.

```
~/OMFIT-source 53 : git add modules/profiles/quantities.txt
~/OMFIT-source 54 : echo $EDITOR
/opt/local/bin/nedit_
```

In (t)csh use `setenv EDITOR <editor>`. In bash use `export EDITOR=<editor>`

10. Use `git commit` to commit your changes. Here I use the `-a` option to commit all changes for all tracked files (use with caution, make sure you really want to commit everything)

```
COMMIT_EDITMSG - /Users/smithsp/OMFIT-source/.git/
File Edit Search Preferences Shell Macro Windows
/Users/smithsp/OMFIT-source/.git/COMMIT_EDITMSG byte 141 of 1081 L: 1 C
1 Give better reporting when the profiles module's EFIT dependency is missing. Convert the 'quantities' entry from
  OMFITtree to OMFITnamelist.]
2
3 <<<>>profiles module<<<>>
4
5 # Please enter the commit message for your changes. Lines starting
6 # with '#' will be ignored, and an empty message aborts the commit.
7 # On branch fix_profiles_dep
8 # Changes to be committed:
9 #   deleted:   modules/profiles/OMFITprofiles/SettingsNamelist.txt
10 #  modified:  modules/profiles/OMFITsave.txt
11 #  modified:  modules/profiles/SCRIPTS/extractProfiles.py
12 #  modified:  modules/profiles/SettingsNamelist.txt
13 #  new file:   modules/profiles/quantities.txt
14 #  deleted:   modules/profiles/quantities/OMFITpickled_-622409702319806493
15 #  deleted:   modules/profiles/quantities/OMFITpickled_1388100350429397418
16 #  deleted:   modules/profiles/quantities/OMFITpickled_2663605247595161454
17 #  deleted:   modules/profiles/quantities/OMFITpickled_6501580777319343817
18 #  deleted:   modules/profiles/quantities/OMFITpickled_8826197606892468825
19 #  deleted:   modules/profiles/quantities/OMFITpickled_896175114144539550
20 #
21
```

The message should be descriptive of what you changed, sometimes why you changed it. The tag in <<<>>tag<<<>> is parsed specially by OMFIT and shows up as blue in the splash screen at startup. Release tags can be added as

<<<\*>>Topic:Subtopic:Description<<<\*>>, such as

<<<\*>>Modules:EFITtime:Added new constraint on q0 as a function of time<<<\*>>

The tag message can also be indicated in the first line of the commit message as  
Topic:Subtopic:Description

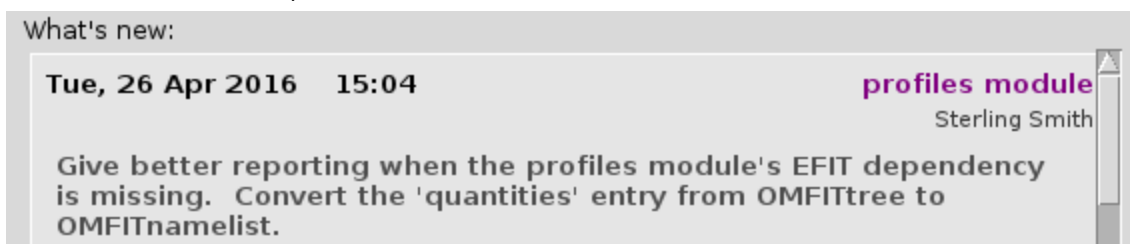
As well the release notes can be indicated in the first line of the commit message with a leading ! as

!Topic:Suptopic:Description



Then close out of the editor. (If you forgot to set your EDITOR, you might be in vi for which you can use `:wq` to get out.)

- a. See [Appendix B](#) for the checks that OMFIT performs when trying to commit code, especially if you get an error when committing.
11. Now your commit message should be first in the What's New screen (when running your own version of OMFIT)



12. If you want to modify your commit message use `git commit --amend`

- According to the agile methodology of programming, there are no bugs, just unwritten regression tests. Whether you are adding a new feature or fixing a bug, you should find a place in an existing regression script (OMFIT-source/regression/test\_\*.py) or add a new script in that directory that tests your addition. Scripts can be run from the command line with `omfit -g <path to script>`. Edit (Sept 2019): Some regression tests are run automatically when pushing your changes.
- Commit your changes to the regression script or whatever fixes were necessary to allow the regression to run (I had to modify one of the scripts)

```
~/OMFIT-source 17 : git commit --amend -a
[fix_profiles_dep b33f85b] Give better reporting when the profiles module's EFIT
dependency is missing. Convert the 'quantities' entry from OMFITtree to OMFITn
amelist.
Date: Tue Apr 26 15:04:05 2016 -0700
12 files changed, 102 insertions(+), 16 deletions(-)
delete mode 100644 modules/profiles/OMFITprofiles/SettingsNamelist.txt
create mode 100644 modules/profiles/quantities.txt
delete mode 100644 modules/profiles/quantities/OMFITpickled_-622409702319806493
delete mode 100644 modules/profiles/quantities/OMFITpickled_1388100350429397418
delete mode 100644 modules/profiles/quantities/OMFITpickled_2663605247595161454
delete mode 100644 modules/profiles/quantities/OMFITpickled_6501580777319343817
delete mode 100644 modules/profiles/quantities/OMFITpickled_8826197606892468825
delete mode 100644 modules/profiles/quantities/OMFITpickled_896175114144539550
~/OMFIT-source 18 : git status
On branch fix_profiles_dep
nothing to commit, working directory clean
```

- Push your changes:

```
~/OMFIT-source 19 : git push -u origin fix_profiles_dep
X11 forwarding request failed on channel 0
Counting objects: 3518, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (1363/1363), done.
Writing objects: 100% (3518/3518), 3.74 MiB | 1.29 MiB/s, done.
Total 3518 (delta 2567), reused 2986 (delta 2107)
To git@github.com:smithsp/OMFIT-source.git
 * [new branch]      fix_profiles_dep -> fix_profiles_dep
Branch fix_profiles_dep set up to track remote branch fix_profiles_dep from orig
in.
```

- Go to GitHub


Your recently pushed branch should show up. Click the Compare & pull request button

- Fill out the pull request with `gafusion/OMFIT-source` as the base fork and `unstable` as the base branch

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base fork: **gafusion/OMFIT-source** base: **unstable** ... head fork: **smithsp/OMFIT-source** compare: **fix\_profiles\_dep** ✔ Able to merge. These branches can be merged.



### profile module robustness

Write Preview AA B i ↺ ↻ ↶ ↷ ↸ ↹

Give better reporting when the profiles module's EFIT dependency is missing. Convert the 'quantities' entry from OMFITtree to OMFITnamelist.

```
<<<>>profiles module<<<>>
```

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported Create pull request

**Labels** ⚙

None yet

---

**Milestone** ⚙

No milestone

---

**Assignee** ⚙

No one—assign yourself

Feel free to modify the title or comment section to be more verbose or request comments from people by @mentioning them. Click the Create pull request button.

This repository Search Pull requests Issues Gist


**gafusion / OMFIT-source** PRIVATE Unwatch 67 Unstar 6 Fork 7

[Code](#) [Issues 158](#) [Pull requests 9](#) [Wiki](#) [Pulse](#) [Graphs](#) [Settings](#)

## profile module robustness #914

✔ Open smithsp wants to merge 1 commit into gafusion:unstable from smithsp:fix\_profiles\_dep

Conversation 0 Commits 1 Files changed 12 +102 -16



smithsp commented just now

Give better reporting when the profiles module's EFIT dependency is missing. Convert the 'quantities' entry from OMFITtree to OMFITnamelist.

This was created as part of creating the 1st commit tutorial.

[Give better reporting when the profiles module's EFIT dependency is m...](#) b33f85b

Add more commits by pushing to the **fix\_profiles\_dep** branch on **smithsp/OMFIT-source**.

✔ **This branch has no conflicts with the base branch**  
Merging can be performed automatically.

✔ Merge pull request You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

**Labels** ⚙

None yet

---

**Milestone** ⚙

No milestone

---

**Assignee** ⚙

No one—assign yourself


---

**Notifications**

Unsubscribe

You're receiving notifications because you authored the thread.

---

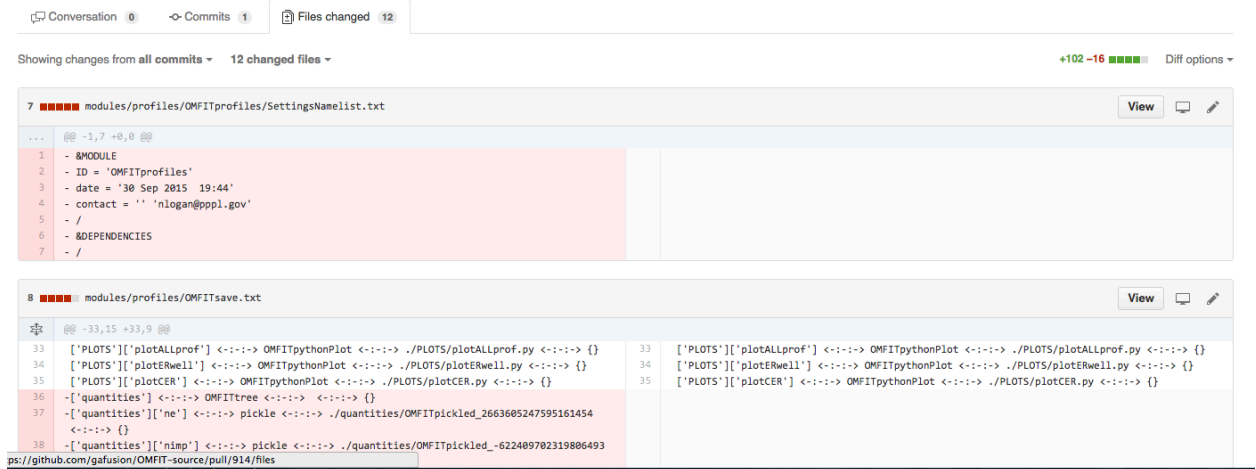
1 participant 

Do not click the “Merge pull request” button. One of the strengths of OMFIT and GitHub is the opportunity for code review. If you merge the pull request right away, you are skipping the review part. If there is a merge that needs to occur between two branches without review, you should use the command line.

As per the comment on that page

**Add more commits by pushing to the `fix_profiles_dep` branch on `smithsp/OMFIT-source`.**

when you add more commits to the same branch that was pushed and for which the pull request was made, these additional commits are just added on to the pull request. So when somebody makes a comment on your request that you should change something, you can make those changes on your branch, test them, commit them, and push them. The files changed tab

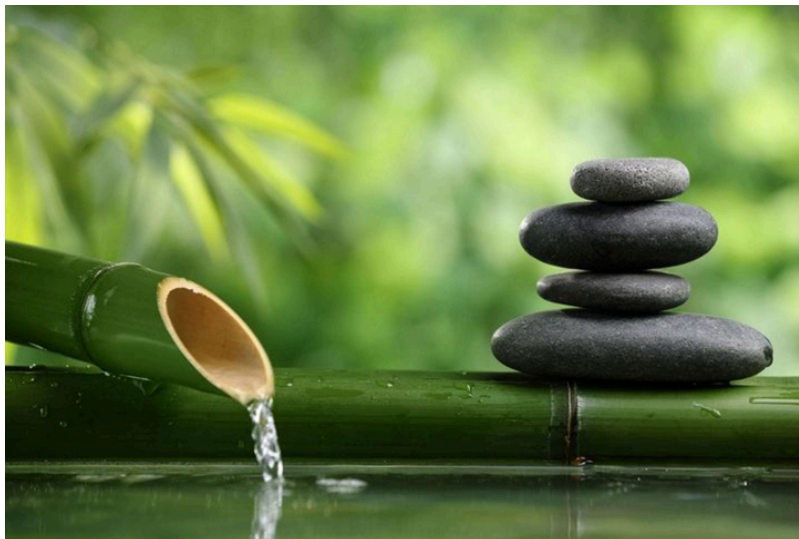


The screenshot shows the 'Files changed' tab of a GitHub pull request. At the top, it indicates 'Showing changes from all commits' and '12 changed files'. Two files are listed:

- 7 modules/profiles/OMFITprofiles/SettingsNameList.txt**: This file has 7 lines. Lines 1-4 are removed (indicated by red background), and lines 5-7 are added (indicated by green background). The content includes metadata like ID, date, and contact, and a section for dependencies.
- 8 modules/profiles/OMFITsave.txt**: This file has 38 lines. Lines 33-35 are removed (red background), and lines 36-38 are added (green background). The content is a JSON-like structure defining plot functions and quantities.

can be useful for seeing what you changed and making line by line comments.

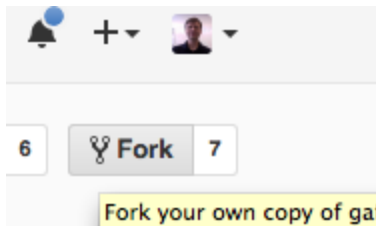
18. Sit back and pat yourself on the back for contributing back to the OMFIT effort!



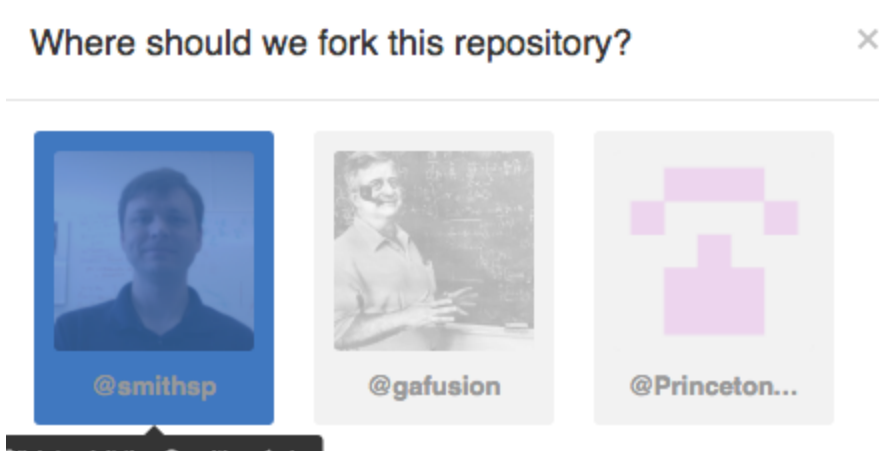
## Appendix A: managing your own fork of the OMFIT-source repository

(These instructions are valid, but be aware that if you choose this path, then any pull requests you open will not be tested by the automatic machinery, so you will not be able to merge to the unstable branch from your fork's branches. You will have to push your branch to the gafusion org to get the automatic machinery working.)

From <https://github.com/gafusion/OMFIT-source/> choose Fork in the upper right hand corner



Fork it to your own area in the pop up window



After you have created your fork, you should delete any unwanted branches from your fork (for instance at <https://github.com/smithsp/OMFIT-source/branches>)

On the machine where you are running OMFIT, in the cloned repository directory, do (replacing the smithsp as appropriate)

```
git remote add smithsp git@github.com:smithsp/OMFIT-source.git
```

In the future, if you want to checkout an existing branch, and that branch exists for two remotes, you will need to be explicit about where to check it out:

```
~/OMFIT-source 126 : git checkout BES
error: pathspec 'BES' did not match any file(s) known to git.
~/OMFIT-source 127 : git branch -a | grep BES
remotes/buchanj/BES
remotes/dmkriete/BES
remotes/logan-nc/BES
remotes/vali/BES
~/OMFIT-source 128 : git checkout -b BES dmkriete/BES
Branch 'BES' set up to track remote branch 'BES' from 'dmkriete'.
Switched to a new branch 'BES'
Cleared temporary .pyc files
```

## Appendix B Commit Checks

### Formatting

At some point in the history of OMFIT, there were debates about how long lines should be, whether to spread function arguments across multiple lines, etc. Then we got the black formatter, and there were no more arguments. OMFIT will not allow you to commit changes that violate the formatting. So you have two options when git commit fails due to formatting issues:

1. Run the formatter and then try to commit again
2. Use the commands:

```
git add list_of_files
```

```
make commit message="Topic:Subtopic:Commit message"
```

to form two commits - the first where the formatting check is skipped, and the second to apply the formatting.

### File size

Git is meant to be used for text based files, not large binary data files. There is a check on the file size, and your commit is rejected if trying to commit too large of a file.

### The use of `OMFIT [` in modules

Scripts contained in a module should not contain absolute references to the OMFIT variable. Instead the scripts should reference `root` to refer to its own module or it should reference a dependency of the module via the `DEPENDENCIES` variables. OMFIT will not allow you to commit code that references `OMFIT [` in a module's scripts.

### Bare exceptions

OMFIT relies on the `KeyboardInterrupt` type of exception to be able to abort an in-progress process. If a script catches that process but continues forward, then the process is not aborted. The script could accidentally catch the `KeyboardInterrupt` if it has a bare

exception statement like

```
except:
```

Instead every exception in OMFIT should specify the exceptions that are expected, such as

```
except ValueError:
```

```
except KeyError:
```

or at the very least

```
except Exception:
```

OMFIT will not allow you to commit code that has the bare exception.

## Module integrity

When you try to commit a change to a module that changes the module structure (adding new file nodes or removing them), OMFIT checks the module structure (the `OMFITsave.txt` file) and verifies that you also added the new file to git as part of the commit. OMFIT will not allow you to commit changes to the module structure that don't include the associated files.