**GitHub Flow based workflow/procedure document**

Github is the centralized repository that will be used.

[www.github.com](http://www.github.com)

The organization is:

[https://github.com/<your\_org\_here](https://github.com/%3cyour_org_here)>

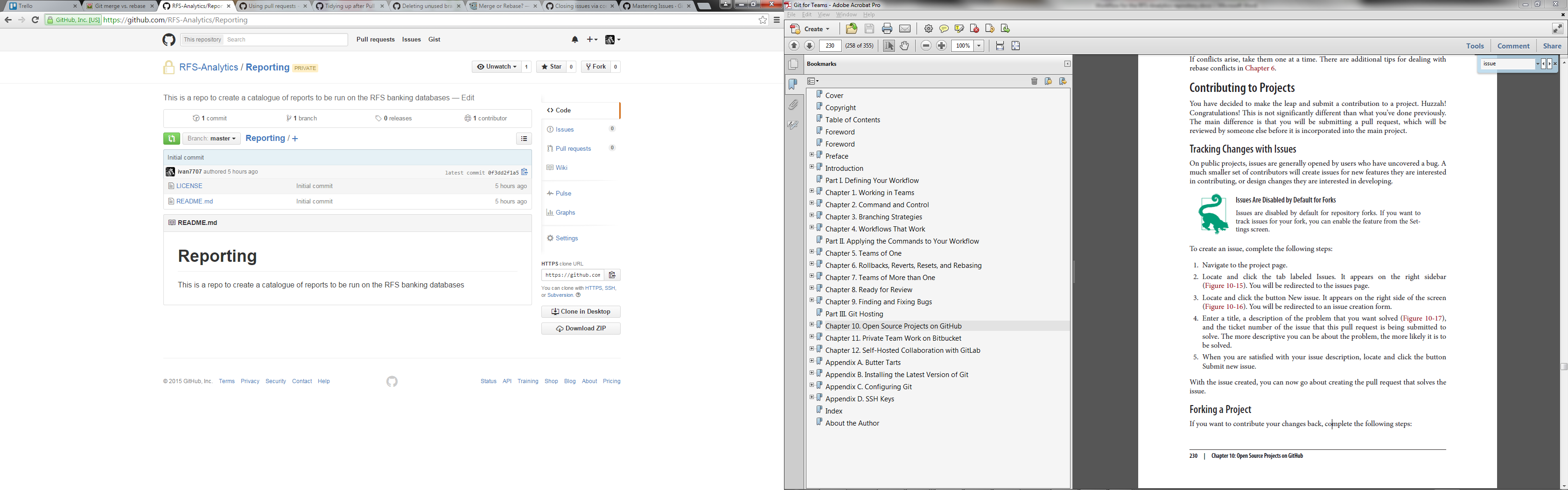
The main reporting repository is:

[https://github.com/<your\_org\_here >/<your\_repo>](https://github.com/%3cyour_org_here%20%3e/%3cyour_repo%3e)

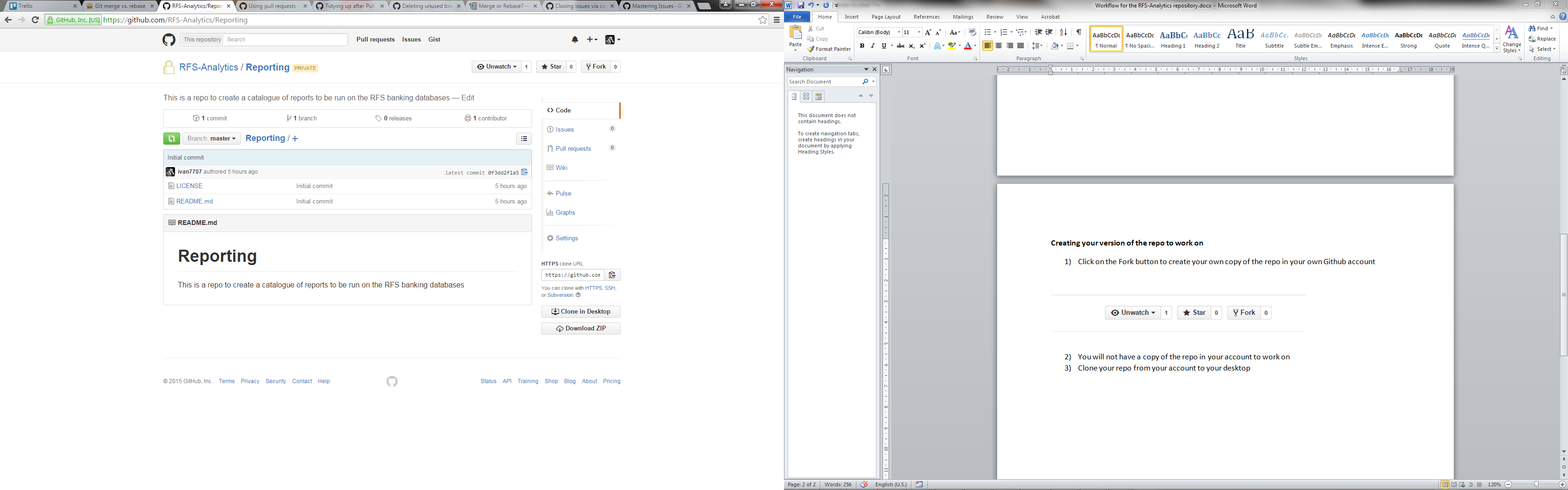
Workflows will all start with creating an issue in GitHub to track the impending change. This way, everyone will be able to keep track of which scripts are being worked on by whom, and what issues in current scripts need to be resolved.

# Creating your version of the repo to work on (ONLY DONE ONCE)

1. Click on the Fork button to create your own copy of the repo in your own Github account



1. You will now have a copy of the repo in your account to work on
2. Clone your repo from your account to your desktop (e.g. git clone <url> <folder> command). Github does have a Windows tool that some may find useful.



1. Add an upstream remote to the main repo (you only have to do this once for your local repo and will be used for keeping your info up to date):
   1. git remote add upstream [https://github.com/<your\_org\_here >/<your\_repo>](https://github.com/%3cyour_org_here%20%3e/%3cyour_repo%3e)

# Adding or Revising the Repo

**New script**

Navigate to the issues page of the main repo:

[[https://github.com/<your\_org\_here >/<your\_repo>](https://github.com/%3cyour_org_here%20%3e/%3cyour_repo%3e)/issues](https://github.com/RFS-Analytics/Reporting/issues)

1. Click on new issue
2. Add a descriptive title and comments about what changes you intend on making
3. Add yourself as assignee on the right hand side
4. Add the appropriate tags (e.g. new script)
5. Click on submit issue

**Enhancement of existing script/bug fix**

1. Navigate to the issues page of the main repo:

[https://github.com/<your\_org\_here >/<your\_repo>/issues](https://github.com/%3cyour_org_here%20%3e/%3cyour_repo%3e/issues)

1. Click on new issue
2. Add a descriptive title and comments about what changes you intend on making
   1. In the body add a link to the script
   2. In the title add the name of the script.
   3. If you are going to fix the issue, add yourself as assignee on the right hand side
3. Add the appropriate tags (e.g. bug fix, enhancement)
4. Click on submit issue

**Making the changes**

**In your forked copy of the repo**

1. Create a branch with a terse description on what the branch is for as the title of the branch
   1. git checkout –b new\_branch\_terse\_desc
2. Complete the revisions/additions to the code to your local copy
3. Ensure your local copy is up to date before pushing (**see Keeping your local and fork up to date**)
4. Push your changes to your forked repo on Github (you will now be ready to create a pull request to merge the changes into the main repo)
   1. Git push origin new\_branch\_terse\_desc

**Keeping your local and fork up to date**

Since there may have been updates to the main repository since you started working on your new code, it is important to get your local branch and forked copy of master current.

At this point you would have already created the new branch and done some work. You should commit that work:

1. git commit –m “message here”

**Update your forked copy of master (on github)**

1. git fetch upstream
2. git checkout master
3. git merge upstream/master
4. git push origin master

**Update the starting point for your local changes**

1. git fetch upstream
2. git checkout feature\_branch
3. git rebase master

If git cannot rebase, you will get merge conflicts that you have to resolve.

**Get the rebase done (one way to fix the merge conflict)**

1. fix the file where the merge conflict occurred. There will be indicators in the file about where the changes are.
2. git add <filename>
3. git rebase –continue

More info: <https://help.github.com/articles/resolving-a-merge-conflict-from-the-command-line/>

**Initiating pull requests**

1. Go to your forked repo on Github
2. Click the compare and review button
3. Base will be ‘master’ (the default branch), as that will be the one main branch with the most up to date status.
4. Click create pull request
   1. be sure to add the issue # that you created when starting the work in the commit message to close the issue (e.g. the changes I provided here closes #5)
   2. closes, close, closed, fix, fixes, resolves are all keywords for closing issues