Git Revision Control

The main Perdigao data system configuration is in the public-field-projects repository on github.com: https://github.com/ncareol/public-field-projects. Despit e the name, this repository is no longer public, so your github account must have been given permission to access this repository.

This configuration is deployed to many places. It is used to run the data processing on *ustar* and at EOL on *barolo*, besides existing on every single DSM in perdigao. So it is important use revision control to keep the configuration consistent and to track all the changes.

This presents some challenges using git on the shared *daq* and *isfs* accounts. The basic tips are available on this software engineering wiki page:

> http://wiki.eol.ucar.edu/sew/GitSharedAccounts

The quick reference section provides enough to get started, but there are lots of details there in case of questions. After logging in with the sshgit script posted on that page, you should be able to commit in *<home>/isfs/projects*, and you will be recorded as the git author. Of course, this assumes that you are set as the git author on the host from which you're logging in. In summary:

From a personal account:

1. Visit the wiki page above and download the sshgit script, and set it executable.
2. Make sure *git config user.name* and *git config user.email* are correct, otherwise set them like so:

```
git config --global user.name "First Last"
git config --global user.email "username@ucar.edu"
```

3. Make sure your current session can authenticate to github:

```
git ls-remote git@github.com:/ncareol/public-field-projects
```

   If not, follow the steps on github.com to create a SSH key and authorize it for your github.com account.
4. Log into the remote shared account with the script:

```
sshgit daq@ops-perdigao.dyndns.org
```

From the shared account

Use git as usual on the shared account:

1. cd isfs/projects/Perdigao
2. git pull
3. git commit
4. After committing, it's a good idea to check the author information on the commit, just in case, with *git show*.
5. git push

The same steps work for logging into the isfs account on barolo. However, for that account, the operational configuration under ~isfs/isfs has group write permissions, so EOL users should be able to pull into that repository while logged into their own account. You can also edit, commit, and push, but we risk interfering if multiple people edit there. Same goes for */net/isf/isfs/projects*.

Initial project creation:

1. cd isfs/projects
2. git add RELAMPAGO
3. git commit RELAMPAGO
4. cd RELAMPAGO
5. git push

Fixing Authorship

If you forget to use the sshgit script to log in, you may still be able to commit to a repository, but the author of the commit will be wrong. You can fix it like so:

```
First log off the remote host.
Log back in with sshgit.
cd to the repository
Fix the last commit: git commit --amend --reset-author
```

The git command will also prompt for a new commit message, but the previous commit message is provided as a default.