Fusion Data Framework Documentation

Release 0.0.0

John Schmitt, David R. Smith, Kevin Tritz, Howard Yuh

October 06, 2015

CONTENTS

CHAPTER

ONE

ABOUT FDF

Fusion Data Framework (FDF) is a data access, management, and visualization framework for magnetic fusion experiments.

Code repository: https://github.com/Fusion-Data-Framework/fdf

HTML Documentation or PDF Documentation

Submit bugs or feature requests: https://github.com/Fusion-Data-Framework/fdf/issues

Created by:

- John Schmitt, Princeton Plasma Physics Lab
- David R. Smith, U. Wisconsin-Madison
- Kevin Tritz, The Johns Hopkins U.
- Howard Yuh, Nova Photonics

To contribute to the FDF project, please contact John, David, Kevin, or Howard.

CHAPTER

TWO

USER GUIDE

This guide is for people who want to use FDF on the PPPL Linux cluster. If you wish to contribute to the FDF code repository (https://github.com/Fusion-Data-Framework/fdf), see the developer guide.

HTML documentation is also available: http://fusion-data-framework.github.io/fdf/

To use FDF on the PPPL Linux cluster (portal.pppl.gov), load the module nstx/fdf (you may need to unload other nstx modules):

```
[sunfire06:~] % module load nstx/fdf

[sunfire06:~] % module list
Currently Loaded Modulefiles:
1) torque/2.5.2      5) idl/8.2           9) java/v1.6
2) moab/5.4.0        6) nstx/treedefs    10) nstx/mdsplus5
3) ppplcluster/1.1   7) nstx/epics       11) nstx/fdf
4) freetds/0.91      8) nstx/idldirs
```

Verify that python points to `/p/fdf/anaconda/bin/python`:

```
[sunfire06:~] % which python
/p/fdf/anaconda/bin/python
```

If python does not point to `/p/fdf/anaconda/bin/python`, then PATH contains to a different python distribution. In this case, you need to modify PATH so `/p/fdf/anaconda/bin` is the first python distribution in PATH.

Finally, you can launch python and import the FDF package:

```
[sunfire06:~] % python
Python 2.7.10 |Anaconda 2.3.0 (64-bit)| (default, Sep 15 2015, 14:50:01)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> import fdf
>>>
```

CHAPTER

THREE

DEVELOPER GUIDE

This guide is for people who want to contribute to the FDF code repository while working within the PPPL Linux cluster environment. If you simply want to use FDF on the PPPL Linux cluster, see the user guide.

The FDF code repository is hosted on GitHub: https://github.com/Fusion-Data-Framework/fdf

To participate in the FDF project as a developer, you must create a GitHub account. The FDF project uses Git for collaborative development, version control, and GitHub communication.

Configure Git

On the Linux cluster at PPPL (portal.pppl.gov), load the module git/1.8.0.2:

```
[sunfire08:~] % module avail git
--------------------- /usr/pppl/Modules/modulefiles -------------------
git/1.7.4.1(default)     git/1.8.0.2      git/2.4.2

[sunfire08:~] % module load git/1.8.0.2

[sunfire08:~] % module list
Currently Loaded Modulefiles:
1) torque/2.5.2      3) ppplcluster/1.1
2) moab/5.4.0        4) git/1.8.0.2
```

On portalr6 (the Red Hat 6 cluster at PPPL) use git/2.4.2. You may want to add the module load command to your shell start-up files: .cshrc for csh/tcsh or .bash_profile for bash.

Next, you must configure Git with your name and email (the same email associated with your GitHub account):

```
[sunfire08:~] % git config --global user.name "John Doe"
[sunfire08:~] % git config --global user.email "JohnDoe@email.com"
```

Also, we recommend setting an editor (e.g. vi, emacs, nedit) for Git comments:

```
[sunfire08:~] % git config --global core.editor nedit
```

You can inspect your Git configuration in the file ~/.gitconfig. For more information about Git configuration, see https://help.github.com/articles/set-up-git/ or https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup

Clone the FDF repository

Git clones repositories into a new directory in your current directory. Now you can clone the FDF repository located at https://github.com/Fusion-Data-Framework/fdf.git to your local directory (you may need to enter your GitHub username and password):

```
[sunfire08:~] % ls -d fdf
ls: fdf: No such file or directory

[sunfire08:~] % git clone https://github.com/Fusion-Data-Framework/fdf.git
Cloning into 'fdf'...
remote: Counting objects: 619, done.
remote: Total 619 (delta 0), reused 0 (delta 0), pack-reused 619
Receiving objects: 100% (619/619), 783.01 KiB, done.
Resolving deltas: 100% (279/279), done.

[sunfire08:~] % ls -d fdf
fdf/
```

The URL for cloning can be found in the right column of the FDF repository page: https://github.com/Fusion-Data-Framework/fdf.

The cloning example above uses the HTTPS method, but cloning via SSH is also feasible: https://help.github.com/articles/set-up-git/#next-steps-authenticating-with-github-from-git

Finally, add your new fdf directory to the `PYTHONPATH` environment variable:

```
[sunfire08:~] % setenv PYTHONPATH ${HOME}/fdf:$PYTHONPATH
[sunfire08:~] % echo $PYTHONPATH
/u/drsmith/fdf:<other directories>
```

You may want to add this action to your shell start-up files: .cshrc for csh/tcsh or .bash_profile bash. In bash, use the export command to set `PYTHONPATH`.

Git workflow for FDF development

1. Create a development branch (here, we call it devbranch) and checkout the new branch:

```
[sunfire08:~] % cd fdf
```

[sunfire08:~/fdf] % git branch * master

[sunfire08:~/fdf] % git branch devbranch

[sunfire08:~/fdf] % git branch devbranch * master

[sunfire08:~/fdf] % git checkout devbranch Switched to branch ‘devbranch’

[sunfire08:~/fdf] % git branch * devbranch master

Devbranch initializes as a copy of master. `git branch` lists branches in your local repository, and the asterisk denotes the active branch. You can switch between local branches with `git checkout <LocalBranchName>`.

2. Push devbranch to the remote FDF repository at GitHub (you may need to enter your GitHub username and password):

```
[sunfire08:~/fdf] % git push origin devbranch
```

Total 0 (delta 0), reused 0 (delta 0) To https://github.com/Fusion-Data-Framework/fdf.git

- [new branch] devbranch -> devbranch

devbranch is now listed in the FDF repository at GitHub. `origin` is the alias for the remote GitHub repository. You can view your remote repositories and aliases with `git remote -v`.

3. Proceed with FDF development within devbranch: commit changes, add/delete files, and push updates to GitHub.

As you complete small tasks, you should commit changes to your local repository with `git commit -a -m '<mymessage>'`. Also, each commit requires a short message describing the changes:

```
[sunfire02:~/fdf] % git commit -a -m 'added dictionary rows in logbook.py'
[devbranch bb6c58a] added dictionary rows in logbook.py
1 file changed, 16 insertions(+), 21 deletions(-)
```

If you do not specify a commit message with -m option, then Git will open your default editor and ask for a commit message (see Configure Git above). The -a option commits all file changes throughout the branch index, not simply your current directory. The branch index is the list of files Git tracks in the branch. `git commit -a` tracks changes to files in the branch index, so you must add new files to the index and remove deleted files from the index. You can view the branch index with `git ls-files`, and you can add new files to the index and remove deleted files from the index with `git add -A`:

```
[sunfire02:~/fdf] % touch temp.py

[sunfire02:~/fdf] % ls temp.py
temp.py

[sunfire02:~/fdf] % git ls-files temp.py

[sunfire02:~/fdf] % git add -A

[sunfire02:~/fdf] % git ls-files temp.py
temp.py
```

Note that temp.py appeared in the index only after the command `git add -A`. Similarly, deleted files stay in the index until the `git add -A` is given.

When you complete a large task, you should "push" changes to the devbranch on GitHub with `git push`:

```
[sunfire05:~/fdf] % git push origin devbranch
Counting objects: 10, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 1.30 KiB, done.
Total 6 (delta 3), reused 0 (delta 0)
To https://github.com/Fusion-Data-Framework/fdf.git
    129c5d9..a166825 devbranch -> devbranch
```

Again, "origin" signifies the branches on the remote GitHub repo.

4. While you are working locally in devbranch, others may be modifying master at GitHub. When you are ready to merge devbranch into master, you should first merge the latest version of master from GitHub into your local devbranch. To retrieve the latest version of master from GitHub, use `git fetch`:

```
[sunfire05:~/fdf] % git fetch origin master
```

From https://github.com/Fusion-Data-Framework/fdf * branch master -> FETCH_HEAD

Next, verify that you are in devbranch and merge origin/master into devbranch:

```
[sunfire08:~/fdf] % git branch
* devbranch
master

[sunfire05:~/fdf] % git merge origin/master
```

Next, push your local devbranch to devbranch on GitHub:

```
[sunfire05:~/fdf] % git push origin devbranch
```

Finally, on the GitHub website, in the devbranch area, submit a *pull request* to pull devbranch into master.

CHAPTER

FOUR

USAGE EXAMPLES

CHAPTER

FIVE

PACKAGE REFERENCE

5.1 Module factory.py

Root module for the FDF package.

Classes

- Machine - root class for the FDF package
- Shot - shot container class
- Logbook - logbook connection class
- Container - diagnostic container class
- Node - mdsplus signal node class
- FdfError - error class for FDF package

Usage:

```
>>> import fdf
>>> nstx = fdf.Machine('nstx')
>>> nstx.s140000.logbook()
>>> nstx.addshots(xp=1048)
>>> nstx.s140000.mpts.plot()
```

5.2 Class Machine

class `factory.`**`Machine`**(*name='nstx'*, *shotlist=[]*, *xp=[]*, *date=[]*)

Factory root class that contains shot objects and MDS access methods.

Basic class initialization is performed as follows: >>>nstx = Machine(name='nstx')

the Machine class contains a model shot object: nstx.s0

shot data can be accessed directly through the Machine class: >>> nstx.s141398 >>> nstx.s141399

alternatively, a list of shot #'s may be provided during initialization: >>>nstx = Machine(name='nstx', shotlist=[141398, 141399])

or added later using the addshot method: >>>nstx.addshot([141398, 141399])

5.3 Class Shot

class `factory.`**`Shot`**(*shot*, *root=None*, *parent=None*)

5.4 Class Logbook

class `factory.`**`Logbook`**(*name='nstx'*, *root=None*)

5.5 Module fdf_signal.py

fdf-signals.py - module containing Signal class

Classes

- Signal - signal class for data objects

Created on Tue Jun 23 2015

@author: hyuh

5.6 Class Signal

class `fdf_signal.`**`Signal`**(***kwargs*)

sig=fdf.Signal(signal_ndarray, units='m/s', axes=['radius','time'], axes_values=[ax1_1Darray, ax2_1Darray], axes_units=['s','cm']

e.g.: mds.Signal(np.arange((20*10)).reshape((10,20)), units='keV', axes=['radius','time'], axes_values=[100+np.arange(10)*5, np.arange(20)*0.1], axes_units=['s','cm'])

or an empty signal: s=mds.Signal() default axes order=[time, space] sig=fdf.Signal(units='m/s', axes=['radius','time'], axes_values=[radiusSignal, timeSignal])

5.7 Module fdf_globals.py

fdf_globals.py contains package-level constants

Created on Thu Jun 18 11:18:16 2015

@author: ktritz

CHAPTER

SIX

LICENSE

The MIT License (MIT)

Copyright (c) 2015 David R. Smith, Kevin Tritz, Howard Yuh

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER

SEVEN

INDICES AND TABLES

- genindex
- modindex
- search

f

F

L

M

S