
Fusion Data Framework Documentation

Release 0.0.0

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

October 06, 2015

CONTENTS

1	About FDF	2
2	User Guide	3
3	Code Reference	4
3.1	Class Machine	4
3.2	Class Shot	4
3.3	Class Logbook	4
3.4	Class Signal	4
4	License	5
5	Indices and tables	6
	Index	7

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

Repository: <https://github.com/Fusion-Data-Framework/fdf>

Documentation: [PDF](#)

ABOUT FDF

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

Repository: <https://github.com/Fusion-Data-Framework/fdf>

Documentation: [HTML](#) or [PDF](#)

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

CODE REFERENCE

3.1 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])`

3.2 Class Shot

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

3.3 Class Logbook

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

3.4 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])`

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.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

L

Logbook (class in factory), [4](#)

M

Machine (class in factory), [4](#)

S

Shot (class in factory), [4](#)

Signal (class in fdf_signal), [4](#)