FEniCS Project Documentation

Release

FEniCS Project Team

Sep 20, 2017

Contents

This is experimental documentation for the FEniCS Project. This version of the documentation on Read the Docs is under development.

CHAPTER 1

Installation

This guide summarises how to install FEniCS. The most reliable way to get started is using the Docker containers (Linux, macOS and Windows).

For installation in high performance computing clusters we recommend always building from source.

From source

To install the Python-only FEniCS packages:

```
pip3 install fenics-ffc --upgrade
```

This will install FFC and its dependencies. It may be necessary to add the flag `--user`.

Todo

Document build from source for DOLFIN.

Debian/Ubuntu packages

FEniCS is available as a package for Debian and Ubuntu[1] in the official repositories. If you are using Ubuntu, we recommend the Ubuntu PPA.

Ubuntu PPA

The Ubuntu Personal Package Archives (PPA) version is the latest release of FEniCS. To install FEniCS from the Ubuntu PPA:

[1] mshr is not available from official Debian and Ubuntu repositories.

```
sudo add-apt-repository ppa:fenics-packages/fenics
sudo apt-get update
```

Official Debian/Ubuntu repositories

The version of FEniCS in the Debian/Ubuntu repositories[1] is not always the most recent FEniCS release. To install FEniCS from the official Debian/Ubuntu repositories:

```
sudo apt-get update
sudo apt-get install fenics
```

Containers/Docker (Linux, macOS and Windows - 64 bit)

A collection of Docker containers for FEniCS are available. To get started, install Docker, and then run

```
docker run -ti -v $(pwd):/home/fenics/shared -w /home/fenics/shared quay.io/
↪fenicsproject/stable:current
```

A helper script is also available. Run the command:

```
curl -s https://get.fenicsproject.org | bash
```

To run the FEniCS Docker image, use the command *fenicsproject run*. For more options and features, see *fenicsproject help*.

For detailed instruction on the Docker containers and background, a see http://fenics-containers.readthedocs.org/en/latest/ for how to run FEniCS inside a container.

Conda (Linux and macOS - 64 bit)

To install the latest FEniCS release from using conda:

```
conda install -c conda-forge fenics
```

To install a development snapshot:

```
conda install -c conda-forge/label/prerelease -c conda-forge fenics
```

The packages are part of conda forge (see https://anaconda.org/conda-forge/fenics), and the recipes are maintained at https://github.com/conda-forge/fenics-feedstock/.

Note: Conda support is experimental and subject to changes.

Development version

Source code for FEniCS is hosted on bitbucket.org:

- DOLFIN

- UFL
- FFC
- FIAT
- Instant
- Dijitso

CHAPTER 2

Getting started

Todo

Getting started guide.

CHAPTER 3

Developers

This notes are for FEniCS developers.

Development workflows

Todo

Describe work flow that is common across projects.

Making releases

These instructions are for developer making release of FEniCS packages.

Releasing packages

Check that all CI systems are green before making a release.

Python packages

These instruction cover:

- FFC
- FIAT
- UFL
- dijitso
- Instant

Making the release

1. Checkout branch and make sure it is clean:

```
git checkout master
git clean -fdx
```

2. Update release notes.
3. Update version number in `<src>/__init__.py`
4. Commit changes.
5. Tag repository and push tag:

```
git tag -a $VERSION -m "Version $VERSION"
git push origin $BRANCH
```

6. Update version number and add `dev` in `<src>/__init__.py`. Commit and push.

Uploading to PyPI

This applies to packages FFC, Instant, Dijitso, FIAT, UFL and the FEniCS Project Python Metapackage.

This should be done soon after a release is made.

1. Checkout release tag and make sure it is clean:

```
git checkout tags/$VERSION
git clean -fdx
```

2. Build source distribution:

```
python3 setup.py sdist
```

3. Sign the package:

```
gpg --detach-sign -a dist/package-1.0.1.tar.gz
```

   The gpg key is in the Steering Council LastPass account.

4. Upload to PyPI:

```
twine upload dist/package-1.0.1.tar.gz dist/package-1.0.1.tar.gz.asc
```

   The username on PyPI is `fenicsproject`.

Note: To use the PyPI test repository:

```
twine upload --repository-url=https://test.pypi.org/legacy/ dist/*
```

Todo

- Update on Read-the-Docs

DOLFIN

Todo

Fill in

Docker containers

Todo

Fill in

Anaconda packages

Todo

Fill in

Ubuntu PPA

Todo

Fill in

Versioning scheme

FEniCS uses a hybrid date-based/serial version numbering scheme. Releases have a version number:

```
year.number.fix
```

In the case of packaging errors (rather than bugs in the release) you can also introduce .postx releases, e.g.:

```
year.number.fix.post0
```

FFC and the FEniCS Python Metapackage

Assume we want to release the `2017.2.0` release. `master` would currently specify:

```
VERSION = "2017.2.0.dev0"
RESTRICT_REQUIREMENT = ">=2017.2.0.dev0,<2017.3"
```

1. On the release branch `2017.2.0`:

```
VERSION = "2017.2.0"
RESTRICT_REQUIREMENTS = ">=2017.2,<2017.3"
```

The upper bound should be *tight* against the current release, i.e. don't do:

```
VERSION = "2017.2.0"
RESTRICT_REQUIREMENTS = ">=2017.2,<2018.1"
```

2. The update to `master` post-release would be:

```
VERSION = "2018.1.0.dev0"
RESTRICT_REQUIREMENTS = ">=2018.1.0.dev0,<2018.2"
```

Other FEniCS Python packages

1. For the release branch, remove the `.dev0` suffix:

```
VERSION = "year.number.0"
```

2. On release of a new version, the `VERSION` string in `setup.py` on `master` branches of the FEniCS Python packages should be incremented:

```
VERSION = "year.number+1.0.dev0"
```

In the case of a release late in the year, it may be more appropriate to increment the year:

```
VERSION = "year+1.number.0.dev0"
```

Todo

Document and explain

CHAPTER 4

Documentation for components

FEniCS is a collection of inter-operating modules. Links to the documentation for each module are listed below. For end-users, the DOLFIN and UFL documentation is most relevant.

- DOLFIN
- mshr
- UFL
- FFC
- FIAT
- Instant
- Dijitso

Documentation build status

Main	
DOLFIN	
FFC	
UFL	
FIAT	
Instant	
Dijitso	