

GNSSRO UFO Hackathon, August 21-27, 2018

Please follow steps 1-8 before attending the GNSSRO code sprint. They provide instructions to build JEDI and run 3D-VAR with the GNSSRO refractivity forward operator that was ported from GSI by Hailing Zhang. This operator will serve as template throughout the code sprint. Once finished, please send us your file *CubedSpherePlot_Field-T_Level-40.png* to acknowledge success of the installation.

Step-by-step guide

0. Forewords

To avoid compatibility and compilation issues, the code sprint will be conducted in the *singularity* container. To expedite the compilation of JEDI, all the needed libraries needed have been installed inside the JCSDA singularity image. Using the image allows each participant to work on his/her own laptop under the same environment (Ubuntu version 16.04). Instructions and pointers on how to install *singularity* are provided in section 1 of this page. Please note that the installation of *singularity* will require the admin/root privileges.

For the participants that do **not** want to install *singularity*, instructions to build JEDI on native unix platforms (linux PC and HPC cheyenne) are available here:

[Building UFO bundle on a Linux PC](#)

[Build JEDI environment on Cheyenne](#)

1. Installing the singularity container

The complete documentation of the *singularity* installation can be found here:

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/developer/jedi_environment/index.html

For Mac users, a step-by-step installation guide is provided below:

[Install Singularity on Mac OS X](#)

In the following, we assume that the singularity container is running in the *vagrant* virtual machine on a Mac OS, and that the JEDI source code will be installed and build in, respectively, the following directories:

`~/gnssro/src`

`~/gnssro/build`

2. Setting up git and access to GitHub

The JEDI code is hosted on GitHub and working with JEDI requires repeated accesses to the site. It is recommended that you create a ".gitconfig" file in your home directory (inside the container if working from a container) with the following content (or, if you already have a ".gitconfig" file, modify it accordingly):

```
$ cat ~/.gitconfig

[user]

    name = Your Name

    email = yourname@somewhere.something

[credential]

    helper = cache --timeout=3600
```

Where "Your Name" and "yourname@somewhere.something" is respectively your GitHub login (user name) and the email address attached to this identifiant.

Making these changes before you build the code will enable you to avoid entering your username and password for every git clone command. Since the bundle accesses many repositories, this can save you time. The last line in the above example ensures that the system will remember your password for a given time (defined in seconds by the timeout parameter).

3. Cloning the JEDI source code

The JEDI full documentation JEDI can be found at:

<https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/index.html>

For the purpose of the code sprint, the various components needed from JEDI to create GNSSRO Unified Forward Operators (UFO) and running 3D-VAR with the FV3 model have been bundled into a single repository named "fv3-bundle". The structure of this directory with its components will be described during the first day of the code sprint. The "fv3-bundle" directory is merely a collection of configuration files, but does not actually contain source code. The source code that makes up the various components of the bundle ("fv3", "ufo", "ioda", etc.) will be automatically cloned during the build process using "git" commands. The installation process, therefore, begins by cloning the bundle from JCSDA's GitHub site:

```
mkdir -p ~/gnssro/src; cd ~/gnssro/src;  
git clone https://github.com/JCSDA/fv3-bundle.git
```

If the JCSDA singularity image is not used, users must also check out the *ecbuild* utility needed to compile and make sure that "ecbuild/bin/ecbuild" is in the path.

```
git clone https://github.com/UCAR/ecbuild.git
```

singularity users don't need to check out "ecbuild", as it is already part of the environment.

4. Selecting the GNSSRO code sprint branch

The JEDI code is in constant flux and a stable branch has been created for purpose of the code sprint. This branch is named "feature/gnssro_code_sprint" and contains the code of the GSI refractivity forward operator and the input data files needed to run 3D-VAR on the FV3 model with that operator. This branch is now frozen, so that it will serve as baseline during the code sprint. Participants are invited to develop on their own branch. The process to create a branch from the existing "feature/gnssro_code_sprint" is described in the next two sections.

```
cd ~/gnssro/src/fv3-bundle  
git checkout feature/gnssro_code_sprint
```

Verify the current branch with the command:

```
git branch -a
```

5. Cloning the code of the various bundle components

The "ecbuild" utility will bring onto the local disk the various components that are listed in file "fv3-bundle/CMakeLists.txt". This is the first part of the build process that precedes compilation. The commands are:

```
mkdir -p ~/gnssro/build/fv3-bundle; cd ~/gnssro/build/fv3-bundle;  
ecbuild ~/gnssro/src/fv3-bundle
```

Note: Before running the "ecbuild" command, make sure to specify:

```
export "FC=mpif90"
```

6. Building the code

Configure the code distribution with the latest changes:

```
mkdir -p ~/gnssro/build/fv3-bundle; cd ~/gnssro/build/fv3-bundle;  
ecbuild ~/gnssro/src/fv3-bundle
```

Compile:

```
make -j4
```

Note 1: The "-j" arguments indicate how many processors to use for the compilation. Your machine may have more or less. Use "-j1" to compile serially.

Note 2: Before running the "ecbuild" command, make sure to specify:

```
export "FC=mpif90"
```

7. Running GNSSRO 3D-VAR with FV3 data

A successful compilation should produce executable "~/gnssro/build/fv3-bundle/bin/fv3jedi_4dvar.x". To run 3D-VAR:

Go to the test directory:

```
cd ~/gnssro/build/fv3-bundle/fv3-jedi/test
```

Run the executable:

```
mpiexec -n 6 ../../bin/fv3jedi_4dvar.x testinput/3dvar_fv3gfs_GnssroRef.yaml
```

Note: the example above execute 6 tasks (-n 6), this is the minimum required to run FV3.

8. Plotting the increment

A successful completion of the 3D-VAR program will end with the error code 0 and message:

```
Run: Finishing oops::Variational<FV3JEDI> with status = 0
```

The analysis increment can be plotted using the PlotCubeSphereField.py graphic utility. To create an image:

Go to the data directory:

```
cd ~/gnssro/build/fv3-bundle/fv3-jedi/test/Data/RESTART
```

Run the Python script:

```
python PlotCubeSphereField.py
```

The analysis increment at level 40 is rendered in file "CubedSpherePlot_Field-T_Level-40.png".

If your singularity DISPLAY is properly set-up, you can visualize the image with:

```
display CubedSpherePlot_Field-T_Level-40.png
```

If not, copy the file to the folder common to the host and virtual machines:

```
cp CubedSpherePlot_Field-T_Level-40.png ~/singularity-vm/vagrant_data/.
```

Visualize the image on your Mac with *preview* or any other graphic applications.

9. Creating your own branch

This task can be performed at the beginning of the code sprint.

It is recommended that each participant create and develop on his/her own branch. This will avoid conflicts within participants during the code sprint. Branches will be merged at the end of the code sprint. There are mainly three components that will be affected by code changes related to GNSSRO new UFOs, those are: "ufo", "ioda" and "fv3-jedi". Participants are free to use any name they like, but the syntax must begin with "feature". For consistency we recommend the following naming convention: "feature/gnssro_code_sprint_<initials>". For example, to create a new branch "gnssro_code_sprint_fcv" in the "ufo" subdirectory:

- 1) Go to the subdirectory and make sure you are starting from the "feature/gnssro_code_sprint" branch:

```
cd ~/gnssro/src/fv3-bundle/ufo
git checkout feature/gnssro_code_sprint
```

- 2) Verify the current branch with the command:

```
git branch -a
```

- 3) Create the new feature branch locally:

```
git branch feature/gnssro_code_sprint_fcv
git checkout feature/gnssro_code_sprint_fcv
```

- 4) Publish the new feature branch to GitHub:

```
git push --set-upstream origin feature/gnssro_code_sprint_fcv
```

Note: Using the "git flow" utility installed in the *singularity* environment, the commands in 3) and 4) reduce to:

```
git flow init -d
git flow feature start gnssro_code_sprint_fcv
git flow feature publish gnssro_code_sprint_fcv
```

Since subdirectories "fv3-bundle/ioda" and "fv3-bundle/fv3-jedi" are also likely to be modified, 1-4 must be repeated in those directories as well.

Finally, the bundle configuration file must be updated with the new branch name. Edit file "~/src/fv3-bundle/CMakeLists.txt" and tell "ecbuild" to check out and build the new branch:

ecbuild_bundle(PROJECT ioda GIT " https://github.com/JCSDA/ioda.git "	BRANCH feature/ gnssro_code_sprint_fcv UPDATE)
ecbuild_bundle(PROJECT ufo GIT " https://github.com/JCSDA/ufo.git "	BRANCH feature/ gnssro_code_sprint_fcv UPDATE)
ecbuild_bundle(PROJECT fv3-jedi GIT " https://github.com/JCSDA/fv3-jedi.git "	BRANCH feature/ gnssro_sprint_fcv UPDATE)

10. Resources

Full JEDI documentation:

<https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/index.html>

JEDI Academy presentations:

[JEDI Academy](#)

To install singularity on Mac OS:

[Install Singularity on Mac OS X](#)

To install singularity on linux:

https://www.sylabs.io/guides/2.5.1/user-guide/quick_start.html#installation

To build JEDI with singularity:

[Build JEDI environment with Singularity](#)

To build JEDI on linux (without singularity):

[Building UFO bundle on a Linux PC](#)

To build JEDI on NCAR HPC cheyenne (without singularity):

[Build JEDI environment on Cheyenne](#)

Understanding and compiling JEDI bundles:

[How to use JEDI bundles](#)

Understanding ecbuild and ctest:

[JEDI Academy ecbuild](#)

Understanding JEDI Fortran interface for UFO implementation:

[Fortran Interfaces for JEDI](#)

Creating a new branch with git:

[JEDI academy git-flow](#)



Related articles

- [Infrastructure Knowledge Base](#)
- [GNSSRO UFO Hackathon, August 21-27, 2018](#)
- [Running HOFX tests on the gnssro branch](#)
- [Build JEDI environment with Singularity](#)
- [Build JEDI environment with Docker](#)