

muffin (branch: **master.python3**) installation HOW-TO

March 8, 2024

This is a brief guide to installing **cGENIE.muffin** (branch: **master.python3**) under **Ubuntu**.

These instructions are valid for a fresh install of **Ubuntu** distribution version 22.04 LTS ('Jammy Jellyfish'). For a different distribution or more established installation, different or fewer respectively components may be needed to be installed and may require a little trial-and-error.

Instructions are given step-by-step, although not all the components need be installed in this order. Note that the various **netCDF** component version numbers may not be the current releases. The most recent versions can almost certainly be substituted (but not tested here) with the caveat that you may not be able to mix-and-match very old with very new libraries.

0.1 Preparation

Get hold of a computer with **Ubuntu** installed on it. Make sure you have plugged in the network cable. Log in. Obtain a strong cup of coffee.

0.2 Installation

1. Get the code!

You may as well start off by cloning¹ the **muffin** code (although you could equally do this last).

From your home directory:

```
git clone https://github.com/derpycode/cgenie.muffin.git
```

Then ... the following installation steps² are in approximately the order you would encounter if you went straight to running the model test (`make testbiogem`) only having cloned the code-base and not accomplished anything particularly constructive.

2. **make**

```
sudo apt install make
```

3. **gfortran** [FORTRAN compiler]

```
sudo apt install gfortran
```

... and then stupid XML support, a 'xml2-config utility' (and the 'libxml2 development package') ... just because ... and then followed by a cascade of library dependencies ...

¹If your system does not know what **git** is (it should be present by default on Ubuntu 22.04): `sudo apt install git`

²They need not be installed in this order (and some may already exist on your system).

4. **xsltproc** [*a command line tool for applying XSLT style-sheets to XML documents*]³

```
sudo apt install xsltproc
```

5. **libxml2-dev**

```
sudo apt install libxml2-dev
```

6. **m4**

```
sudo apt install m4
```

7. **libcurl4-openssl-dev**

```
sudo apt install libcurl4-openssl-dev
```

8. **libz-dev**

```
sudo apt install libz-dev
```

9. **libhdf5-dev**

```
sudo apt install libhdf5-dev
```

(If your system cannot find the **libhdf5-dev** package, try: `sudo apt update -y` first)

Now(!) you are ready for **netCDF**. These libraries come rather inconveniently in multiple parts ... first we need to install the main **netCDF C** libraries and then the **FORTRAN** libraries that depend on the **C** libraries^{4 5 6}.

10. **netcdf-c** [netCDF C libraries]

```
wget https://downloads.unidata.ucar.edu/netcdf-c/4.9.2/netcdf-c-4.9.2.tar.gz
tar xzf netcdf-c-4.9.2.tar.gz
cd netcdf-c-4.9.2
./configure
```

At this point, you may well see: *'configure: error: Can't find or link to the hdf5 library. Use -disable-hdf5, or see config.log for errors'* because the **hdf5** libraries you have just only just installed, mysteriously cannot be located ... You need to add their paths, e.g., for Ubuntu 22.04:

```
export LDFLAGS="-L/usr/lib/x86_64-linux-gnu/hdf5/serial/lib"
export CFLAGS="-I/usr/lib/x86_64-linux-gnu/hdf5/serial/include"
```

(If you need to find where **hdf5** is hiding: `dpkg -L libhdf5-dev`).

Repeat `./configure` if necessary and continue:

```
make check
sudo make install
```

```
(cd ..)
```

For the next step, it can be that the libraries you have just installed cannot be 'found'. You can force an update of the library link cache by:

```
sudo ldconfig
```

³In the **cupcake** release we do not need this :)

⁴The examples given are for the most recent versions of the libraries. For details/most recent version, see: <https://www.unidata.ucar.edu/software/netcdf/>

⁵(From some random convenient download/temporary directory.)

⁶In **master.python3**, you no longer need the **C++** libraries :o)

11. **netcdf-fortran** [netCDF FORTRAN libraries]

```
wget https://downloads.unidata.ucar.edu/ ...  
... netcdf-fortran/4.6.1/netcdf-fortran-4.6.1.tar.gz  
tar xzf netcdf-fortran-4.6.1.tar.gz  
cd netcdf-fortran-4.6.1  
./configure  
make check  
sudo make install  
  
(cd ..)
```

That is is for the basic installation!

0.3 Testing muffin

First, from **cgenie.muffin** make sure that you are on the correct branch:

```
git checkout master.python3
```

To test the code installation – change directory to **cgenie.muffin/genie-main** and type:

```
make testbiogem
```

This compiles a carbon cycle enabled configuration of **muffin** and runs a short test, comparing the results against those of a pre-run experiment (also downloaded alongside the model source code). It serves to check that you have the software environment correctly configured.

If when running **make testbiogem** you run into issues, specifically: libraries that cannot be 'found', try forcing an update of the library link cache (the not found libraries may be links to the 'real' library and somehow this link is not working/found):

```
sudo ldconfig
```

If you are still unsuccessful at this point ... double-check the software and directory environment settings in **user.mak** (or **user.sh**) and for a **netCDF** error, check the value of the **NETCDF_DIR** environment variable. (Refer to the FAQ section for addition fault-finding tips.) If environment variables are changed: before re-trying the test, you will need to type:

```
make cleanall
```
