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

March 11, 2024

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

These instructions are valid for a fresh install of MAC OS intel and M1/M2 chip distributions. 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 **MacOS** installed on it. Make sure you have plugged in the network cable. Log in. Obtain a strong cup of coffee.

0.2 Installation

1. First of all, you will need **XCode**, which can be downloaded from the app store, or here:
   https://developer.apple.com/xcode/ (and then -¿ download)
   After installing **XCode**, it is necessary to enable command line tools, by entering at the command line:

   ```
   xcode-select --install
   ```

2. Get **Homebrew** by pasting the following – ALL AS ONE LINE — at the terminal command line:

   ```
   /usr/bin/ruby -e /bin/bash -c
      "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
   ```

   Next, type:

   ```
   brew doctor
   ```

   This should tell you "your system is ready to brew".[1]

3. Install **Fortran**, **C++**, **netCDF**, and some other useful libraries at the command line (using **Homebrew**) as follows:

   ```
   brew install cmake
   brew install gcc
   brew install hdf5
   brew install netcdf
   brew install netcdf-fortran
   brew install netcdf-cxx
   brew install wget
   ```

[1] If it doesn't, see Section 21.6.

4. Check your **netCDF** version number and path to the installed libraries (you will need this info shortly ...) by entering:

   ```
   brew info netcdf
   ```

   This will return several lines, but the key one gives the **netCDF** path and should look like something like[2]:

   ```
   /usr/local/Cellar/netcdf/4.9.2_1 (84 files, 6.2MB)
   ```

5. Get hold of a current copy of the muffin code:

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

That is is for the basic installation!

0.3 Configuration

6. From the directory cgenie.muffin, switch to a code branch that can accommodate all types of Apple silicon:

   ```
   git branch master.python3
   ```

7. To configure your installation, change to the directory: cgenie.muffin/genie-main and then edit the file: user.mak as follows:

   - Under:
     ```
     # === Machine type (LINUX/SLOARIS/SGI) ===
     ```
     comment out (add a #) the default setting
     ```
     MACHINE=LINUX
     ```
     and then un-comment (remove the #) from one of the following 2 lines, depending on whether you have an older (intel processor) or newer (M1 or M2 chip) Mac:
     ```
     #MACHINE=OSX # Intel processor
     #MACHINE=OSX_M # Apple silicon (M1, M2 etc.)
     ```
   - At the end of user.mak, comment out the default **netCDF** library setting, and then un-comment the line:
     ```
     #NETCDF_DIR=/usr/local/Cellar/netcdf/4.9.2_1
     ```
     and edit this setting if your **netCDF** library location is different (see above).
     If you find yourself experiencing vexing **netCDF** library feeling, you can also cross your fingers and try:
     ```
     NETCDF_DIR=/usr/local
     ```

8. Finally, to test the code installation – from cgenie.muffin/genie-main 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 you get **netCDF** related errors, try running:

   ```
   make cleanall
   ```

   first and then re-try make testbiogem.

---

[2] For the example of a install with version **4.9.2_1**