

Setup Guide for a Linux Development Environment

Nick Macey

March 16, 2021

1. Assumptions

1.1 Libraries, and executables built manually, should be placed in a `/prod` directory. Access to this should be controlled, but can (and should) be a group distinct to the sudoers.

1.2 The instructions are for a Debian-like distro (Ubuntu, Mint). Broadly, they'll work for any distro, but note that anything using `apt` will be different.

2. LLVM

2.1 LLVM version 11.0 onwards includes the flang (Fortran) compiler.

2.2 Unfortunately, LLVM 11.0 does not exist in any useful pre-built binary or package file.

2.3 Easiest way to build and install LLVM 11.0 is to use the following script:

- https://www.dropbox.com/s/87115qvebssjack/build_llvm.sh?dl=0
- Will install into `/prod/bin`, `/prod/lib` and `/prod/include`
- Note: Building LLVM requires a recent copy of both `gcc` and `gfortran`

3. HDF5

3.1 Pre-built binaries are only available for CentOS – not terribly helpful given that CentOS is dead...

3.2 Download source code from here:

- <https://www.hdfgroup.org/downloads/hdf5/source-code/>
- Ideally, you want the Cmake version (bottom of the page)

3.3 Modify the HDF5options.cmake configuration file (NOT the HDF5config.cmake file)

- Switch between MPI and serial builds
- Switch on Fortran build
- Modify install prefix

3.4 Export compiler executables. E.g. for GCC:

- `export CC=gcc CXX=g++ FC=gfortran`

3.5 Build via cmake and make (or `cmake --build`)

3.6 Note that HDF5 v1.12 does not seem to work with Clang

3.7 Export HDF5_DIR as described here: <https://portal.hdfgroup.org/display/support/CMake+Scripts+for+Building+Application>

4. SILO

4.1 SILO is probably the most reliable mesh output format for a general Linux install, although it does suffer from a really bad Fortran API (so we'll install NetCDF too).

4.2 If you want to use HDF5 with SILO, you'll need an older HDF5 library (1.10). But SILO works perfectly well with its own file format.

4.3 Download from here: <https://wci.llnl.gov/simulation/computer-codes/silo> and follow the build instructions in the INSTALL file.

5. NetCDF

5.1 Download from here:

- <https://www.unidata.ucar.edu/downloads/netcdf/>
- You want both the C and Fortran libraries, and possibly the C++ one

5.2 Build using Cmake, following these guides:

- https://www.unidata.ucar.edu/software/netcdf/docs/getting_and_building_netcdf.html#netCDF-CMake
- https://www.unidata.ucar.edu/software/netcdf/docs/building_netcdf_fortran.html
- Remember to set the installation directory to /prod, not the default of /usr/local

6. Module Environment

6.1 Download from <https://sourceforge.net/projects/modules/files/Modules/>

6.2 Build instructions are here: <https://modules.readthedocs.io/en/stable/INSTALL.html>

- You'll need tclsh installed (`sudo apt install tcl tcl-dev`)
- Probably does make sense to install this into /usr/share (see example at bottom of installation guide)
- Make sure to follow the instructions in the Configuration section in order to have the module environment load

7. AMReX

7.1 Clone AMReX repository: `https://github.com/AMReX-Codes/amrex`

7.2 Create BUILD directory

7.3 Run cmake

- `cmake -DAMReX_OMP=YES -DAMReX_FORTRAN_INTERFACES=YES -
DAMReX_HDF5=YES -DAMReX_SPACEDIM=2 -DCMAKE_INSTALL_PREFIX=/
prod/AMReX2D_CMAKE ..`
- (Requires that `HDF5_ROOT` be set to the root of a parallel HDF5 build)

7.4 Make and install

- `make; make install`