Building UFO bundle on a Linux PC

For those wishing to build and run the UFO bundle on a standalone Linux PC outside of a container

Step-by-step guide

1. Download ecbuild from: https://github.com/ecmwf/ecbuild
2. Install it in a place mentioned in $PATH
3. Clone the ufo bundle with your github username: **% git clone https://<username>@github.com/JCSDA/ufo-bundle.git**
4. As of this writing (7/17/2018), an example build environment for Linux PC exists only in a feature branch, though hopefully it'll be merged to the development branch soon. For now, switch branches like this after cd-ing into the ufo bundle: **% git checkout feature/buildtools**
5. cd into the **tools/** directory and edit the file **module_setup_linuxpc_gcc_openmpi.sh** The naming of this file indicates that it is for a Linux PC using the GCC compiler suite and openmpi MPI library. You'll need to ensure that gcc, gcc-fortran, and openmpi are installed before proceeding. If you wish to use a different MPI stack that shouldn't be too hard, but may require some changes to, and renaming of, the module_setup* file.
6. The first 6 "export" entries in module_setup_linuxpc_gcc_openmpi.sh indicate installation locations for packages required by the UFO bundle. The following steps outline what I did for a successful build on my Ubuntu/LinuxMint PC:
    a. For eigen: **% sudo apt-get install libeigen3-dev**
    b. For netcdf: **% sudo apt-get install libnetcdf-c++4-dev; sudo apt-get install libnetcdff-dev**
    c. For CXX_COMPILER and C_COMPILER and Fortran_COMPILER: **%sudo apt-get install gcc-5-base; sudo apt-get install libgfortran5**
    d. For boost: I was not able to get a package install approach to work, so instead built it from scratch. Rev. 1.67.* or greater should work. I won't kid you: getting this monster to build and install correctly is not simple. Unfortunately I didn't take notes on what I did to get the install to work, or I'd have included them here.
7. Still in the **tools/** directory, run the script **build.sh** This script was originally designed to simplify the build process on a few known machines, and has been augmented here for a generic Linux PC. The script takes 4 mandatory arguments and one optional. The first is hostname (**linuxpc**), the second is compiler suite (**gcc**), the third is MPI distribution (for me this was **openmpi**), the fourth can be either **debug** or **release.** The fifth optional argument is the number of threads to allow "make" to use. This can speed up the build process substantially. I generally set it to the number of processors available on the machine on which the build is being done. So, the following works for me:
   % build.sh linuxpc gcc openmpi release 4
8. After a successful build, you can run the full test suite by doing the following:
    a. **% source module_setup_linuxpc_gcc_openmpi.sh** (For csh users: % source module_setup_linuxpc_gcc_openmpi.csh)
    b. Run the test suite: **% cd ..; ctest**

Related articles

- Building UFO bundle on a Linux PC
- Build JEDI environment on Cheyenne