How to build HOMME with CMake

Building HOMME requires working MPI Fortran and C compilers, Netcdf built with these or the serial version of these compilers, and CMake.

Building HOMME with CMake

To build HOMME using the CMake build system, please follow these steps:

1) Check out HOMME

Check out the source code

```
svn co https://svn-homme-model.cgd.ucar.edu/trunk homme-trunk
```

2) Create a location (not the source directory) in which to build HOMME and enter that directory.

Make a directory for the build

```
mkdir /scratch/user_name/my-homme-build
cd /scratch/user_name/my-homme-build
```

3) Create a script called config.sh to hold the configure specific information. Copy the following text into the script and change the compiler names and paths for your system. Note the line continuation characters at the end of each line (there cannot be additional whitespace after this character). Additionally, the last line should always be a path to the appropriate HOMME checkout.

config.sh

```
#!/bin/bash

rm -rf CMakeFiles CMakeCache.txt

cmake \
  -DCMAKE_Fortran_COMPILER=ftn \
  -DCMAKE_C_COMPILER=cc \
  -DCMAKE_CXX_COMPILER=CC \
  -DNETCDF_DIR:PATH=/opt/cray/netcdf/4.0.1.3/netcdf-pgi \
  -DPNETCDF_DIR:PATH=/contrib/pnetcdf/1.2.0/pgi \
  -DHDF5_DIR:PATH=/opt/cray/hdf5/1.8.4.1/hdf5-pgi \
  /path/to/homme-trunk
```

Be sure to set the executable permission on config.sh.

4) Run ./config.sh (ensure that the script has executable permissions set properly). If this passes successfully, continue to step 5. If config.sh failed, determine the cause of the failure and ammend config.sh. For example if you receive the message

Possible CMake configure error

```
CMake Error at cmake/FindNetcdf.cmake:167 (MESSAGE):
  SZIP Not found, set SZIP_DIR to appropriate installation
Call Stack (most recent call first):
  CMakeLists.txt:112 (FIND_PACKAGE)
```

then you need to specify the location of the missing package in config.sh. Here we would add

Specify SZIP Path

```

-DSZIP_DIR:PATH=/home/research/user_name/software/szip-2.1 \
```

to the config.sh file. Re-run the script, specifying the location of other packages (if needed) until the build is successfully configured.

The full dependency list of HOMME is: netcdf,pnetcdf,hdf5,curl,szip,zlib. However, not all of these are required since different builds of Netcdf and HDF5 may have different dependencies. Additionally, CMake may be able to find some of the dependencies in the system libraries so you may not need to explicitly set them as shown above. It is recommended that you begin specifying the path to Netcdf, Pnetcdf, and HDF5 as shown in the example config.sh and then add the other libraries as needed.

5) Once successfully configured, the code can be compiled with make. The HOMME CMake build handles highly parallelized builds; to speed up the compilation use "make -j #" where # is the number of processes to use for compilation.

Compiling HOMME

```
make -j 4
```

To compile or recompile only one executable or build target you can simply specify the name of the target when using the make command. Note that because of dependencies other targets may get built/rebuilt as well. To view the build targets available you can use the "make help" command.

View build targets

```
And to compile a single target from this list you simply run equivalent of the following.
```

Compiling a single target

```
make -j 4 preqx
```

6) If make succeeds you have successfully built HOMME. The executables preqx, sweqx, and swdgx are located in src/preqx/, src/sweqx/, and src/swdgx/ respectively.

Locate the executables

```
user_name@localhost:homme-build> ls src/preqx/preqx
src/preqx/preqx*
user_name@localhost:homme-build> ls src/sweqx/sweqx
src/sweqx/sweqx*
user_name@localhost:homme-build> ls src/swdgx/swdgx
src/swdgx/swdgx*
```

Additional note for building on supercomputers

Because most supercomputers use a queuing system (such as pbs or lsf) to manage parallel runs, running HOMME on such systems adds additional challenges. In particular, instead of running the job directly from the command line, a submission file must be prepared and submitted to the queue. These submission files typically include information about the job including the number of nodes and processes required, the maximum time of the run, an account charge number, and the command to run the parallel executable. Since HOMME is typically run on supercomputers with queuing, the setup of submission files has been integrated into the HOMME configure system.

A configuration of HOMME for machines with a queuing system requires more information from the user. In particular, the charge number and the type of queue must be supplied. The charge number can either be set as an environment variable (HOMME_PROJID) or as an argument to the cmake command at configure time.

```
-DHOMME_PROJID=XXXX0000
```

The type of queuing is automatically determined to be LSF or PBS if the commands bsub or qsub, respectively, are available.