CMAQv5.2 Quickstart Guide

06/30/2017

0.1 CMAQ Installation Quick Start Guide

0.1.1 System Checks

The following support software are required for compiling and running CMAQ.

1. Fortran and C compilers, e.g., Intel, Portland Group, Gnu
2. Git
3. I/O API
4. netCDF
5. Message Passing Interface (MPI), e.g., OpenMPI or MVAPICH2.

0.1.2 Install CMAQ and Required Libraries

In the directory where you would like to install CMAQ, create the directory issue the following command to clone the EPA GitHub repository for CMAQv5.2:

```
git clone -b 5.2 https://github.com/USEPA/CMAQ.git CMAQ_REPO
```

For instructions on installing CMAQ from tarballs, see Chapter 5.

0.1.3 Check Out A new Branch in the CMAQ Repository

Checking out a new branch is a good idea even if you are not doing code development, per se. It is likely that you will want to retrieve new updates in the future, and an easy way to do this is through the master branch in the git repo. Thus it is beneficial to leave it unperturbed if possible.

```
cd CMAQ_REPO
git checkout -b my_branch
```

0.1.4 Configure the CMAQ build environment

The user has two options for building an environment. She or he may build and run CMAQ components directly in the repository structure (object files and executables will be ignored with .gitignore), or they may extract the build and run scripts out of the repository and work in a separate location. If you would like to build directly in the repository, skip to "Install the CMAQ Libraries" below.

0.1.4.1 Build and run in a user-specified directory outside of the repository

In the top level of CMAQ_REPO, the bldit_project.csh script will automatically replicate the CMAQ folder structure and copy every build and run script out of the repository so that you may modify them freely without version control.

In bldit_project.csh, modify the variable $CMAQ_HOME to identify the folder that you would like to install the CMAQ package under. For example:

```
set CMAQ_HOME = /home/username/CMAQ_v5.2
```

Now execute the script.

```
./bldit_project.csh
```

0.1.5 Install the CMAQ Libraries

The CMAQ build scripts require the following libraries and INCLUDE files to be available in the CMAQ_LIB directory (Note the CMAQ_LIB gets set automatically by the config_cmaq.csh script, where `CMAQ_LIB = $CMAQ_HOME/lib`):

- netCDF library and INCLUDE files are located in the `$CMAQ_LIB/netcdf` directory
- I/O API library and module files are located in the `$CMAQ_LIB/ioapi` directory
- MPI library and INCLUDE files are located in the `$CMAQ_LIB/mpi` directory

The config_cmaq.csh script will automatically link the required libraries into the CMAQ_LIB directory. Set the locations of the netCDF, I/O API, and MPI installations on your Linux system with the following config_cmaq.csh environment variables:

- `setenv IOAPI_MOD`: the location of the I/O API module files on your system.
- `setenv IOAPI_INCL`: the location of the I/O API include files on your system.
- `setenv IOAPI_LIB`: the location of compiled I/O API libraries on your system.
- `setenv NETCDF`: the location of the netCDF installation on your system.
- `setenv MPI`: the location of the MPI (OpenMPI or MVAPICH) on your system.

For example, if your netCDF libraries and includes files are installed in /usr/local/netcdf, set `NETCDF` to /usr/local/netcdf. Similarly, if your I/O API library is installed in /home/cmaq/ioapi/Linux2_x86_64ifort, set `IOAPI_LIB` to /home/cmaq/ioapi/Linux2_x86_64ifort.

1. Check the names of the I/O API and netCDF libraries using the `ioapi_lib` and `netcdf_lib` script variables.

2. Check the name of the MPI library using the `mpi` script variable. For MVAPICH use `-lmpich`; for openMPI use `-lmpi`.

Links to these libraries will automatically be created when you run any of the build or run scripts. To manually create these libraries (this is optional), execute the config_cmaq.csh script, identifying the compiler in the command line [intel | gcc | pgi]:

```
./config_cmaq.csh [compiler]
```

You may also identify the version of the compiler if you wish it to be identified in build directory and executable names. This is optional. For example:

```
./config_cmaq.csh intel 17.0
```

0.1.6 Compiling CMAQ

Create the model executables for ICON, BCON, and CCTM:

```
cd $CMAQ_HOME/PREP/icon/scripts
./bldit.icon |& tee bldit.icon.log

cd $CMAQ_HOME/PREP/bcon/scripts
./bldit.bcon [compiler]

cd $CMAQ_HOME
./bldit_cctm.csh [compiler]
```

0.1.7 Install the CMAQ input reference/benchmark data

Download the CMAQ reference data from the CMAS Center Software Clearinghouse and copy to `$CMAQ_HOME`. Navigate to the `$CMAQ_HOME` directory, unzip and untar the `CMAQv5.2.DATA.tar.gz` file:

```
cd $CMAQ_HOME
tar xvzf CMAQv5.2.DATA.tar.gz
```

0.1.8 Configure the CCTM script for MPI

For an MPI configuration with 6 processors,

```
cd $CMAQ_HOME
```

Edit the CCTM run script (run_cctm.csh) for the MPI configuration that you will use:

```
setenv NPROCS 6
setenv NPCOL_NPROW "3 2"
```

Most clustered multiprocessor systems require a command to start the MPI run-time environment. The default CCTM run script uses the *mpirun* command. Consult your system administrator to find out how to invoke MPI when running multiprocessor applications.

For single-processor computing, set NPROCS to 1 and NPCOL_NPROW to “1 1”

For single-processor computing,

```
setenv NPROCS 1
setenv NPCOL_NPROW to "1 1"
```

After configuring the MPI settings for your Linux system, using the following command to run the CCTM. Per the note above, different Linux systems have different requirements for submitting MPI jobs. The command below is an example of how to submit the CCTM run script and may differ depending on the MPI requirements of your Linux system.

```
./run_cctm.csh |& tee cctm.log
```