

2. Overview of CMAQ System Components

CMAQ is a suite of Fortran 90 programs that work in concert to estimate ozone, PM, toxic compounds, and acid deposition throughout the troposphere. The five main CMAQ programs are

- The initial conditions processor [ICON](#)
- The boundary conditions processor [BCON](#)
- The clear-sky photolysis rate calculator [JPROC](#)
- The Meteorology-Chemistry Interface Processor [MCIP](#)
- The CMAQ Chemistry-Transport Model [CCTM](#)

Ancillary support programs distributed with CMAQ include

- The code builder/manager [Bldmake](#)
- The chemical mechanism compiler [CHEMMECH](#)
- EBI chemistry solver builder [CREATE_EBI](#)

The following sections describe the CMAQ system concept, followed by [details of the programs listed above](#).

Installation overview

Prior to CMAQ version 5.0.2, CMAQ developers used [CVS](#) for source code management, and distributed [tarballs](#) (except for MCIP) were CVS archives. Starting with version 5.0.2, CMAQ developers switched to [git](#).

All versions of CMAQ from 4.7.1 to the present are available on the [U.S. EPA GitHub repository](#).

CMAQ source codes are used to build "executables": binary files that consist of instructions that have been translated from their original [source code](#) (e.g., Fortran) into [machine code](#) (also called machine language or object code) so that they are ready to be run (executed). Executable files are created through the use of a specialized program called a [compiler](#).

Installing from GitHub

There are two options for obtaining CMAQ source code from GitHub.

1. Download a zipped code archive from the GitHub web client
2. Clone the repository directly to a Linux server using the command line. For example, to download CMAQ version 5.1, issue the following command (this assumes that git is installed on your system):

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

Installing from git-based tarballs (CMAQ version 5.0.2 and later)

Users need not install git to install CMAQ. Zip archives of the source code and scripts are available from Git Hub. Click the "Clone or download" button on the [U.S. EPA GitHub repository](#) and select "Download ZIP" to get a zip file of the full CMAQ repository, including scripts and source code. Unzip this file on the Linux file system directory

where you would like to install CMAQ.

Installing from CVS-based tarballs (CMAQ version 5.0.1 and earlier)

CVS must be installed on the user's Linux system before installing CMAQ versions earlier than 5.0.2. When the distributed CMAQ tar file is unpacked, a CVS directory tree is installed on the user's machine that contains archived copies of the CMAQ source code. The CMAQ program Bldmake controls the extraction of copies of CMAQ source code from CVS based on the configuration options specified by the user in C-shell build scripts. After exporting the CMAQ source code from CVS, Bldmake then invokes a Fortran 90 compiler to compile the CMAQ source code into binary object files and link them with the necessary precompiled libraries to create binary CMAQ executables. C and Fortran 90 compilers must be installed on the user's Linux system in order to create CMAQ executables.

MCIP was not distributed in a CVS archive. Instead, when older CMAQ versions are downloaded, MCIP appears in its own directory with source code and Makefiles for installation.

Configuration options

Because the model infrastructure was designed to promote modularity, the user must create new CMAQ executables for each suite of science configuration options for all programs except MCIP. There are too many combinations of the various chemical mechanisms, horizontal and vertical transport schemes, cloud routines, and chemistry solvers in the CMAQ science configuration to include efficiently in a single executable. The requirement to recompile CMAQ with each science configuration change is offset by the flexibility to add new science to the model or to

switch between different model configurations. This point about modularity is most pertinent to CCTM, although there are configuration options that must be selected when compiling the other CMAQ programs as well.

In addition to compile-time configuration options with CMAQ, there are also execution-time configuration options (options that are chosen when the model is run versus when it is compiled). The horizontal domain configuration and the vertical coordinate system are dynamic features in CMAQ that are independent of the executable. In other words, a user can employ a single executable for a simulation that uses any of the supported map projections or grid definitions, without having to recompile the source code into a new executable. A description of which CMAQ options must be selected at compilation versus at execution are included in [Chapter 7: CMAQ Programs and Libraries](#).

Chemistry-transport model conceptual formulation

As the chemistry-transport model (CTM) component of CMAQ, CCTM is the final program to be run in the CMAQ modeling sequence. There are four other main programs that prepare input data for CCTM (i.e., ICON, BCON, JPROC, and MCIP). Before describing each of the CMAQ programs in the [summary description section](#), we present a conceptual formulation of CMAQ and Eulerian air quality modeling to provide a framework for understanding the purposes of the various programs and their relationships to each other and to the overall system.

Eulerian CTMs use coupled ordinary differential equations (ODEs) to predict changes in pollutant concentrations throughout a three-dimensional grid that is fixed in space. The following processes affect

changes in the predicted concentrations in each grid cell:

- Emissions from sources
- Horizontal and vertical advection
- Horizontal and vertical diffusion
- Chemical transformations
- Loss processes (deposition)

Mathematically, these processes relate to the concentration change in each grid cell over time ($\partial C/\partial t$) through the *continuity equation*, which is presented in simplified form below:

$$\partial C/\partial t = Adv + Diff + Rc + Ec - Sc$$

where,

Adv = advection

Diff = diffusion

Rc = chemical transformation of species c

Ec = emissions of species c

Sc = loss processes for species c

In CMAQ, the advection and emissions terms are calculated based on input files generated by the meteorology and emissions models, respectively; the diffusion, chemical transformation, and loss process terms are calculated within CCTM.

The Eulerian representation of the area to be modeled is a series of contiguous grid cells that form a limited-area modeling domain on a subset of the globe. A limited-area domain requires that boundary conditions be established to account for advection of pollutants and other chemical species into the modeling domain from areas outside it. CMAQ currently accounts for advection into the domain only from the horizontal (i.e., lateral) boundaries, assuming there is no exchange

through the top boundary of the domain (i.e., vertical exchange). These spatial lateral boundary conditions are estimated in CMAQ using the boundary conditions processor, BCON. Similarly, a temporal boundary condition is established with the initial conditions processor, ICON, which estimates the chemical conditions in the first time step of a CMAQ model simulation. To model incoming solar radiation, which provides the energy source for photolysis reactions, the program JPROC calculates clear-sky photolysis rates at various latitude bands and hours based on solar hour angles. Output from these three CMAQ programs is used with output files from the emissions and meteorological models and other CMAQ preprocessors to form the required input data for running CCTM.

Inline processes

In the context of air quality modeling, inline processes refer to those processes that are run at the same time as the chemistry-transport model. The major advantage of including processes inline to the CCTM is that the simulation of one parameter can feedback to the simulation of one or more other parameters. For example, by including the photolysis rate calculations inline in the CCTM, the rates will be influenced by the radiative impacts of the simulated aerosol loading. Other inline features to the CCTM, such as emissions processing, provide efficiency advantages to the modeling operation and facilitate coupled meteorology-chemistry modeling. CMAQ version 5.0 and forward supports emissions calculations for biogenic sources, windblown dust, seasalt, and point source plume rise in the CCTM.

User's must be careful to avoid double counting emissions when designing an inline CCTM simulation. The CCTM does not have the ability to identify which emissions sources were calculated offline and are being input to the model. For example, including biogenic emissions in the input emissions files and configuring the CCTM to calculate the

biogenic emissions inline is redundant and will produce erroneous double counting of the emissions source.

Photolysis rates may also be calculated inline in the CCTM. **When the CCTM is configuring to calculate photolysis rates inline, the preprocessing program JPROC is no longer a required part of the CMAQ modeling sequence.**

Summary descriptions of the major CMAQ programs

The major CMAQ components and ancillary programs are described below. More detailed discussions on the formulations of the CMAQ programs are available in [Chapter 4](#), in Byun and Ching (1999), and in Byun and Schere (2006).

Note that the following list of programs is generally the order in which the CMAQ programs are run.

Model Builder (Bldmake)

Bldmake provides an interface to CMAQ source code repository, and to the Fortran 90 compiler for building binary executables. Because Bldmake is required to create all of the CMAQ executables except MCIP (which has its own Makefile procedure), it is the first program that needs to be compiled after installing the CMAQ source code on your system. In addition to creating executables, it also provides the option to generate a Linux Makefile. These are particularly useful for porting the CMAQ code to new operating systems, testing new code in a development environment, or trouble-shooting problems with CMAQ compilation or execution.

Photolysis Rate Processor (JPROC)

JPROC calculates chemical-mechanism-specific clear-sky photolysis rates at fixed altitudes, solar hour angles, and latitude bands from tabulated absorption cross section and quantum yield (CSQY) data. CMAQ is distributed with CSQY data that support the default chemical mechanisms. The only configuration option required for JPROC is the selection of the chemical mechanism to use in the modeling. Output from JPROC is an ASCII look-up table of photolysis rates that CCTM uses to calculate gas-phase chemical transformations and pollutant concentrations.

JPROC is not required if the CCTM is configured to run with inline photolysis rate calculations.

Initial Conditions Processor (ICON)

ICON generates a gridded binary netCDF file of the chemical conditions in the modeling domain for the first hour of a simulation. It can generate these initial conditions from either an ASCII file of vertically resolved concentration profiles (distributed with CMAQ) or from an existing CCTM output file. If the profiles in an ASCII file do not have the same vertical structure as the CCTM configuration to be used, ICON will interpolate the data to a vertical structure consistent with CCTM's. Using an existing CCTM output file to generate initial conditions is applicable when extrapolating initial conditions from a coarse to a fine grid simulation, as may occur when setting up nested simulations (simulations with finer-resolution grids that cover part of coarser-resolution grids). As discussed in [Chapter 7](#), the configuration options for ICON include selecting the chemical mechanism to model, defining the horizontal and vertical grids, and choosing whether the initial conditions are generated from an ASCII profile or from an existing CCTM output file.

Boundary Conditions Processor (BCON)

BCON generates a gridded binary netCDF file of the chemical conditions along the horizontal boundaries of the modeling domain. These boundary conditions can be either static or time-varying, and (as with ICON) can be generated from either an ASCII file of vertically resolved concentration profiles or from an existing CCTM output file. Also as with ICON, BCON will interpolate the data in ASCII profiles to a vertical resolution that is consistent with the CCTM configuration. BCON differs from ICON, however, in that it can generate time-varying (i.e., dynamic) boundary conditions. Dynamic boundary conditions are typically extracted either from CCTM outputs from a coarse-grid simulation for nested simulations or from a CCTM simulation using a global-scale model. The file structure of the ASCII input profiles can also support the creation of dynamic boundary conditions, but generally these files are used only for creating static data. The configuration options for BCON include selecting the chemical mechanism to model, defining the horizontal and vertical grids, and choosing whether the boundary conditions are generated from an ASCII profile or from an existing CCTM output file. (discussed further in [Chapter 7](#))

BCON is only used to create boundary conditions inputs for the CCTM from an ASCII profile file or from an existing CCTM output file. Users are responsible for preparing CCTM input boundary conditions from other sources, such as global chemistry transport models.

Meteorology-Chemistry Interface Processor (MCIP)

MCIP uses output files from the MM5 or WRF meteorological models to create netCDF-formatted input meteorology data that are used by SMOKE (the emissions processor that computes emissions inputs to CMAQ) and by CMAQ. MCIP prepares and diagnoses all meteorological

fields that are required for SMOKE and CCTM. In addition, MCIP is currently used to calculate the time-varying, species-dependent dry deposition velocities that are used in CCTM. MCIP can be used to uniformly trim cells off the horizontal boundary of the domain defined by the meteorological model, or to window in on a subset of that domain. MCIP can also decrease the vertical resolution of the meteorological data by “layer collapsing,” although this option should be used with caution as it can degrade the quality of the data if used incorrectly. Configuration options for MCIP include the time periods over which to extract data from the meteorological model output files, horizontal and vertical grid definitions, and selections for calculating dry deposition velocities and integrating satellite cloud observations into MCIP output. (discussed further in [Chapter 7](#))

CMAQ Chemistry-Transport Model (CCTM)

CCTM integrates the output from the preprocessing programs described above (JPROC, BCON, ICON, and MCIP), as well as CMAQ-ready emissions inputs (e.g., output from SMOKE), to simulate continuous atmospheric chemical conditions. The modeled concentrations of relevant species can be captured for output at a user-defined time frequency (typically hourly). The CCTM output files are all binary netCDF files of gridded and temporally resolved air pollutant information, such as gas- and aerosol-phase species mixing ratios, hourly wet and dry deposition values, visibility metrics, and integral-averaged concentrations.

The spatial and temporal coverages of CCTM are dictated by the input meteorology information. The science configuration is specific to each application of the model and can be adjusted to optimize model performance both computationally and in the numerical reproduction of observed air quality trends. Configuration options for CCTM include the

temporal coverage of the simulation, the chemical mechanism to use in the modeling, the physics scheme to use for modeling pollutant transport, heterogeneous and aqueous chemistry options, inline processing options, and diagnostic options (such as process analysis, discussed in the next paragraph). CCTM has the largest number of configuration options of all the CMAQ programs. (discussed further in [Chapter 7](#))

Chemical Mechanism Compiler (CHEMMECH)

This program creates chemical mechanism namelist files for CMAQ from a mechanism definition file. Chemical mechanisms are represented in CMAQ through a series of namelist files that contain mechanistic and kinetic parameters that describe a photochemical mechanism.

CHEMMECH creates the namelist files from an ASCII mechanism-definition file that represents the chemistry as sequential equations of reactants, products, and reaction rate information. This program is needed to modify reaction stoichiometry or kinetics in the existing mechanisms, to add new species and reactions, and to implement entirely new chemical mechanisms in CMAQ.

EBI chemistry solver builder (CREATE_EBI)

The Euler Backward Iterative (EBI) chemistry solver is an optimized numerical solver for CCTM photochemical mechanisms. As the EBI solver is optimized for a specific chemistry mechanism configuration, a new version of the EBI solver is required for new CCTM photochemical mechanisms. The program CREATE_EBI is a CCTM source code generator for new mechanism versions. Mechanism input files for CREATE_EBI are produced by the CMAQ program CHEMMECH. The source code generated by CREATE_EBI may be used to compile a new version of the CCTM for use with updated chemistry namelist files

created with CHEMMECH.

Lightning Data Preprocessor (LTNG_2D_DATA)

This program reads lightning flash count data from the National Lightning Detection Network (NLDN) and creates a CCTM input file to use for simulations that include lightning NO_x emissions.

[<< Previous Chapter](#) - [Home](#) - [Next Chapter >>](#)

CMAQ Operational Guidance Document (c) 2016