June 2004

User's Guide to the NCAR Community Atmosphere Model (CAM 3.0)

James R. McCaa
Mathew Rothstein
Brian E. Eaton
James M. Rosinski
Erik Kluzek
Mariana Vertenstein

Climate And Global Dynamics Division
National Center For Atmospheric Research
Boulder, Colorado, USA

Climate And Global Dynamics Division
National Center For Atmospheric Research
Boulder, Colorado

The National Center for Atmospheric Research (NCAR) is operated by the University Corporation for Atmospheric Research (UCAR) and is sponsored by the National Science Foundation. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Figures

List of Tables

Acknowledgments

The authors wish to acknowledge the many members of NCAR's Climate Modeling Section (CMS) and Computer Software and Engineering Group (CSEG) who have contributed to this document over the years.

The new model would not exist without the significant input from members of the CCSM Atmospheric Model Working Group AMWG too numerous to mention. Bill Collins (NCAR), Leo Donner (GFDL), James Hack (NCAR), David Randall (Colorado State University), and Phil Rasch (NCAR) were the co-chairs of the AMWG during the development of CAM 3.0.

We would also like to acknowledge the substantial contributions to the CAM 3.0 effort from the National Science Foundation, Department of Energy, the National Oceanic and Atmospheric Administration, and the National Aeronautics and Space Administration.

Chapter 1

Introduction

The Community Atmosphere Model (CAM 3.0) is the latest in a series of global atmosphere models developed at the National Center for Atmospheric Research (NCAR). This guide is intended to instruct the novice user on running CAM 3.0 and to inform the more experienced user on the changes that have gone into CAM 3.0 from CAM 2.0.

CAM 3.0 contains significant scientific and software changes from the last major release (CAM 2.0). These are discussed below in section 1.3. Please see the CAM 3.0 Scientific Descriptions [Collins et al., 2004] for a detailed discussion of the science in the model. Note that the climate produced by this version of the model is considerably different than that of the CAM 2.0 release.

1.1 How to Use This Document

If you're just anxious to get a model run started and want to sort out the details later, jump to the section named **Quick Start** (1.2).

The novice user should learn how to obtain the source code and datasets (see section 1.4) and then go to **Building and Running CAM** (2) and read the details of how to build and run the model. The novice user would also do well to experiment with the different use cases presented in **Use Cases** (2.3) before they go on to their own work. The glossary will also be useful for the novice user to understand the terms and abbreviations used in this guide.

An experienced user of CAM should read Section 1.3, describing changes since CAM 2.0, and may wish to to reference the various tables on namelist items, output fields, etc.

A user who also needs to make code changes will want to study the DRAFT Interface to Column Physics and Chemistry Packages, which gives some guidance on making changes to the model. When the CAM 3.0 Developer's Guide is complete, it will give more detail on the model internals and how the user would approach making more extensive changes. An experienced user who has also developed their own build/run mechanism may also be interested in the details of the configuration files given in the appendix.

If you have the pdf or html versions of this document, please note the use of hyperlinks in the presented examples – in many cases you can click on namelist variables or command line option and go right to a description of the item in the relevant table.

Lastly, throughout the document `this font is used to indicate shell commands and options, fragments of code, namelist variables, etc.` Where examples from an inter-

active C shell session are presented, lines starting with `%` indicate the shell prompt.

1.2 Quick Start

The following list of steps allows a user to quickly make a one day test run of CAM 3.0.

1. Download the CAM source code, resolution-independent datasets, and T42 datasets from: http://www.ccsm.ucar.edu/php/ccsmAgreement.php?indexId=36. You will need to accept the CAM License Agreement.
2. Untar each dataset:
   ```
   % gunzip -c cam3.0_source_code.tar.gz | tar xvf -
   % gunzip -c cam3.0_forall_datasets.tar.gz | tar xvf -
   % gunzip -c cam3.0_64x128_T42_datasets.tar.gz | tar xvf -
   ```
3. Set the environment variable "CSMDATA" to the location of the input datasets:
   ```
   % setenv CSMDATA `pwd`
   ```
4. Execute the configure command:
   ```
   % ./cam1/models/atm/cam/bld/configure
   ```
5. Build cam:
   ```
   % gmake
   ```
6. Execute the build-namelist command:
   ```
   % ./cam1/models/atm/cam/bld/build-namelist
   ```
7. Run cam:
   ```
   % ./cam < namelist
   ```

At this point CAM should run for one day and exit. For more sophisticated runs you will surely want to read the sections titled **Use Cases** (2.3) and **Sample Run Scripts** (2.4).

1.3 Changes since CAM 2.0

Highlights of the software improvements include:

- Performance improvements for all three dynamical cores in the areas of:
  - load balancing
  - interprocess communication
  - physics and dynamics decompositions
  - compiler optimizations
  - memory requirements
  - vectorization
- Inclusion of a single column model (SCAM).
- New run-time controls for performance and science options.
- Ability to output a contiguous subset of columns to the history tape instead of a full field.

New science developments include:

- Updated parameterizations for prognostic cloud water, cloud ice, precipitation, and cloud fraction.
- Inclusion of a Slab Ocean Model (SOM).
- Inclusion of ISCCP cloud simulator to simulate ISCCP statistical cloud diagnostics.
- Diagnostic treatment of aerosols (sulfate, dust, sea salt, carbon, and volcanic).
- Optional prognostic treatment of sulfate aerosols.
- Improved energy conservation.
- Modifications to longwave interaction with water vapor.
- Updates to shortwave scheme for trace gas absorption
- Atmosphere-land interface now supports rain and snow phases.

1.4 Obtaining and unpacking the source code and datasets

The primary distribution mechanism for CAM is the via the
CAM web page (http://www.ccsm.ucar.edu/models/atm-cam).

Once there, you must follow the download link and accept the CAM distribution license.

At the least you must download the "CAM 3.0 Source distribution", the "Resolution-independent datasets", and the dataset tarball for at least one resolution.

To untar the source code tar file you execute the following command:

```
gunzip -c cam3.0_source_code.tar.gz | tar xvf -
```

To untar one or more dataset tarfiles you must first go a directory that will be the dataset root directory. For all of the examples in this document it will be assumed that the environment variable CSMDATA will be set to the full pathname of this directory. You can achieve this in an interactive shell by typing the following command while in the dataset root directory:

```
setenv CSMDATA `pwd`
```

Now unpack each of the dataset tar files using:

```
gunzip -c <tarfilename> | tar xvf -
```

Note: All dataset tarfiles should be unpacked in the same directory.

1.4.1 CAM input dataset directory hierarchy

The directory structure of the input datasets is as follows:

Table 1.1: **CAM Input Dataset Directory Hierarchy**

Directory name	Synopsis
`inputdata`	Top level directory where CSMDATA points to
`inputdata/atm`	Datasets for atmosphereic models
`inputdata/atm/cam2`	Datasets specifically for CAM
`inputdata/atm/cam2/ggas`	Greenhouse gas forcing datasets
`inputdata/atm/cam2/inic`	Atmosphereic initial condition datasets
`inputdata/atm/cam2/ozone`	Ozone datasets

Directory name	Synopsis
inputdata/atm/cam2/rad	Radiation datasets
inputdata/atm/cam2/scam	Single column model datasets
inputdata/atm/cam2/scyc	Sulfate forcing datasets
inputdata/atm/cam2/sst	Sea Surface temperature datasets
inputdata/lnd	Datasets for land models
inputdata/lnd/clm2	Datasets for CLM
inputdata/lnd/clm2/pftdata	Plant physiology datasets
inputdata/lnd/clm2/rawdata	High resolution surface datasets
inputdata/lnd/clm2/srfdata/cam	Time-invariant surface datasets
inputdata/lnd/clm2/inidata_2.1/cam	Land-model initial condition datasets

1.4.2 CAM source code directory hierarchy

The directory hierarchy for CAM 3.0 is as follows. The directory hierarchy closely parallels the directory hierarchy for CCSM3 and as such seperates code out by model component.

Table 1.2: **CAM Source Code Directory Hierarchy**

Directory name	Synopsis
models/atm/cam/bld/	Scripts to build and execute the model
models/atm/cam/src/	Atmosphere model main source code directory
models/atm/cam/src/advection/slt	Semi-Lagrangian Transport advection routines
models/atm/cam/src/control	Control code
models/atm/cam/src/dynamics/eul	Eulerian dynamics
models/atm/cam/src/dynamics/fv	Finite-Volume dynamics
models/atm/cam/src/dynamics/sld	Semi-Lagrangian dynamics
models/atm/cam/src/ocnsice/dom	Data Ocean Model
models/atm/cam/src/ocnsice/som	Slab Ocean Model
models/atm/cam/src/physics/cam1	Physics routines (e.g., radiation, convection)
models/atm/cam/src/utils	CAM specific utilities
models/atm/cam/tools/	Directory of tools (such as history compare routines)
models/atm/cam/tools/cprnc	History file comparison program. Normally used to compare code modifications to a "base-line" code. Prints out summary of differences.
models/atm/cam/tools/scam	Single column model

Directory name	Synopsis
`models/csm_share/shr`	Code shared by all the geophysical model components of the Community Climate System Model (CCSM) (e.g. code for CCSM message passing and orbital calculations)
`models/lnd/clm2`	Community Land Model (CLM2.1) code
`models/ice/csim4`	Community Sea-Ice Model (CSIM4) code
`models/utils`	Independent utility codes
`models/utils/esmf`	General purpose Earth System Modeling Framework (ESMF) utilities.
`models/utils/pilgrim`	Parallel Library for Grid Manipulations.
`models/utils/timing`	General purpose timing library

1.5 Getting Help – Other User Resources

1.5.1 The CAM Web Page

The central source for information on CAM is the
CAM web page (http://www.ccsm.ucar.edu/models/atm-cam). Here you can find model updates, bug reports, the latest documentation, and much, much more. Visit today!

1.5.2 The cam-users mailing list

The cam-users discussion group is an open e-mail forum for rapid exchange of information, ideas, and topics of interest relating to the various versions of the NCAR CAM. This includes sharing software tools, datasets, programming tips and examples, as well as discussions of questions, problems and workarounds. The primary motivation for the establishment of this list is to facilitate and encourage communication between the users of the CAM around the world. This mail group will also be used to distribute announcements related to the CAM.

To subscribe to this user group go to http://mailman.ucar.edu/mailman/listinfo/cam-users.

1.5.3 Reporting bugs

If a user should encounter bugs in the code (i.e., it doesn't behave in a way in which the documentation says it should), the problem should be reported electronically to **The cam-users mailing list** (1.5.2). When reporting a suspected bug, please include the following information: 1) the architecture on which the code was built; 2) the configuration cache xml file; 3) the namelist input; and 4) Model printout. Please note that that CAM 3.0 is a research model, and not all features are supported.

Chapter 2

Building and Running CAM

This chapter describes how to build and run the standalone model CAM 3.0. This includes a description of the build procedure, a variety of example use cases demonstrating available options, and a discussion of running the executable. CAM 3.0 may also be run as part of the Community Climate System Model (CCSM); this is discussed in Section 2.7 on page 35.

Building and running CAM takes place in the following steps:

1. Configure
2. Build model
3. Build namelist
4. Execute model

Configure includes setting the compile-time parameters such as resolution, dynamical core (Eulerian Spectral, Semi-Lagrangian Spectral, or Finite-Volume), type of parallelism to employ (shared-memory and/or distributed memory), number of constituents, and number of vertical levels. This step is done most easily by invoking the `configure` script that creates the files necissary for the build step to take place. The `configure` utility is discussed in the `?? (??)` section.

Build model includes compiling and linking the executable using the GNU make utility. The configure script creates a copy of the Makefile in the directory where the build is to take place. The user then need only change to this directory and execute `gmake`.

Build namelist includes executing the `build-namelist` script, which supports a variety of options to control the run-time behavior of the model. The `build-namelist` utility is discussed in the `The build-namelist utility (2.2)` section.

Execute model includes the actual invocation of the executable. When running using distributed memory parallelism this step requires knowledge of how your machine invokes MPI executables. When using shared-memory parallelism using Open-MP you may also set the number of Open-MP threads.

It is assumed that the user has access to the utilities `tar`, `gunzip`, `gmake` (GNU `make`), and `perl`. The scripts written in perl need at least Perl 5.4 to work. Lastly, we assume there is a CAM 3.0distribution (with root `$CAM_ROOT`) available.

The most basic execution procedure is:

```
$CAM_ROOT/models/atm/cam/bld/configure
gmake
```

```
$CAM_ROOT/models/atm/cam/bld/build-namelist
./cam < namelist
```

Most users will encapsulate these steps in higher level `perl` or shell scripts. Sample scripts to use as templates for this purpose are included in the directory `$CAM_ROOT/models/atm/cam/bld`. These are discussed in more detail in the section named **Sample Run Scripts** (2.4).

2.1 The `configure` utility

The `configure` utility provides a flexible way to specify a particular configuration of CAM. By default it will produce the configuration files required to build the standard production version of CAM (currently Eulerian dynamics at T42 spectral resolution with 26 levels).

`configure` produces the configuration build files *Filepath*, *misc.h*, *params.h*, *preproc.h*, and *Makefile*. Each of these files specify compile-time parameters and settings needed to build the model. In addition, a configuration cache file (*config_cache.xml* by default) is written which may be used in a subsequent invocation of `configure` to exactly reproduce the configuration files. The files produced by running `configure` are written to the directory where CAM will be built, which by default is the directory from which `configure` is executed, but can be specified to be elsewhere (see the `-cam_bld` option).

`configure` will optionally perform tests to validate that the Fortran compiler is operational and Fortran 90 compliant, and that the linker can resolve references to required external libraries (NetCDF and possibly MPI). These tests will point out problems with the user environment in a way that is much easier to understand than looking at the output from a failed build of CAM. We strongly recommend that the first time CAM is built on any new machine, `configure` should be invoked to execute these tests (see the `-test` option).

2.1.1 Options to configure

All configuration options can be specified in the following ways, listed in order of decreasing precedence:

- by invoking `configure` in interactive prompting mode (enabled with the `-i` option),
- by setting specific options on the command-line,
- by a default configuration cache file (specified using the `-defaults` option).

At the next level of precedence a few options can be specified by setting environment variables. And finally, at the lowest precedence, many options have hard-coded defaults. Most of these are located in the file *config_cache_defaults.xml* in the configuration script directory. A few that depend on previous settings are hard-coded in `configure` (a perl script). The hard-coded defaults will produce the standard production configuration of CAM.

The interactive prompting mode has two levels: basic and expert. The basic mode, which is enabled by the `-i` option, asks the user all questions required to configure CAM, assuming that the model is built entirely from code that is contained in the distribution. The expert mode, which is enabled by setting the `-x` option in addition to `-i`, allows the user to specify that various pieces of code required to build CAM may come from directories outside the distribution. All

the flexibility available in the expert interactive mode is also available from the specific options set on the command-line or from a user specified cache file.

The configure script allows the user to specify compile time options such as model resolution, dynamical core type, additional compiler flags, and many other aspects. The user can type `configure --help` for a complete list of available options.

The following options may all be specified with either one or two leading dashes, e.g., `-help` or `--help`. Options that can be expressed as single letter switches may not be clumped, e.g., `-i -x` may NOT be expressed as `-ix`. When multiple options are listed separated by a vertical bar (|), either version may be used.

Table 2.1: **Command line arguments to configure**

Option Name	**Description**	**Default**
`-cache file`	file specifies the pathname of the output configuration cache file. This file is not used in the build process (i.e., the Makefile does not depend on it), but is used instead to archive a complete description of the configuration of a CAM executable. This file is used by the build-namelist utility for setting default namelist values that depend on the configuration of the CAM executable (e.g., which dynamics package and what resolution are used).	config_cache.xml
`-cam_bld dir`	dir is the directory where CAM will be built. The configuration build files produced by invoking configure are written to this directory. It will eventually contain all the .o and .mod files produced by the build.	directory from which configure is invoked

Option Name	Description	Default
`-cam_cfg dir`	dir is the directory that contains the CAM configuration scripts, which includes perl modules as well as various defaults files that are required by configure. It is possible (but not recommended) to move configure to another directory, but then the configuration script directory must be explicitly specified so that configure can find its support files. Normally this directory will be determined by looking at the pathname that is used to invoke configure (assuming that configure has not been moved from the configuration script directory). If configure is not in the configuration script directory, then that directory can be specified either by this option, by setting the environment variable CAM_CFGDIR to the configuration script directory, or by setting the environment variable `CAM_ROOT` to the root directory of the CAM distribution assuming that the configuration directory is `$CAM_ROOT/models/atm/cam/bld`.	directory part of the absolute pathname used to invoke configure
`-cam_exe name`	name is the name of the CAM executable file.	cam
`-cam_exedir dir`	dir is the directory where the CAM executable will be created.	the CAM build directory
`-cam_root dir`	dir specifies the top level directory of a CAM distribution. This directory must contain the subdirectory models which must contain the subdirectories atm, csm_share, ice, and lnd. The CAM root directory can also be specified by setting the environment variable CAM_ROOT.	config_dir/../../../.. where config_dir is the directory containing the `configure` executable.
`-cc name`	name specifies the C compiler. This allows the user to override the default setting in the Makefile (Linux only). The C compiler can also be specified by setting the environment variable USER_CC.	pgcc if using pgf90, otherwise use cc
`-debug`	Enable the compiler debugging options that are specified in the Makefile.	no debugging

Option Name	Description	Default
`-defaults file`	file specifies the pathname of a configuration cache file that will used to provide default values, e.g. a config_cache.xml from a previous invocation of configure. Note that by default the current configuration will be written to a file named config_cache.xml, so the user should avoid running `configure` in the same directory in which the defaults file is located. Alternatively the output cache file can be renamed by using the -cache option.	none
`-dyn name`	name specifies the dynamics package to be used when running CAM. The valid options are eul (Eulerian spectral dynamics), sld (semi-Lagrangian spectral dynamics), and fv (finite-volume dynamics).	eul
`-esmf_bld dir`	dir is the top level directory where the ESMF library will be built. It will contain the directories mod, lib/libg and/or lib/libO which each contain subdirectories that contain machine architecture specific files.	cam_bld/esmf, where cam_bld is the build directory for CAM.
`-esmf_root dir`	dir is the top level directory for the ESMF distribution. This directory contains the main makefile for the ESMF library. The ESMF root directory can also be specified by setting the environment variable ESMF_ROOT. Note that the ESMF library is supplied as part of the CAM distribution, and by default the supplied library is built and used by CAM. This option allows for a custom build of CAM that builds an ESMF library using source code from outside the CAM distribution.	$CAM_ROOT/ models/utils/esmf, where cam_root is the root directory of the CAM distribution.
`-fc name`	name specifies the Fortran compiler. This allows the user to override the default setting in the Makefile. The Fortran compiler can also be specified by setting the environment variable USER_FC.	OS dependent
`-fflags string`	The Fortran compiler options specified by string will be appended to the default setting of FFLAGS in the Makefile. If string contains any whitespace it must be quoted.	
`-gmake name`	Name of the GNU make program on your system. Supply the absolute pathname if the program is not in your path (or fix your path). This is only needed if testing will be done (see -test).	The names 'gmake', 'gnumake', and 'make' are checked in order.

Option Name	Description	Default
`-h | -help`	Print usage to STDOUT.	
`-i | -interactive`	Turns on the basic interactive prompting mode. In the basic prompt mode only questions required to obtain a standard configuration of CAM will be asked. To allow configurations which require code from outside the CAM distribution, an expert prompt mode can be enabled by additionally specifying the -x option. In either mode a default value for each setting will be determined based on, in order of decreasing precedence, the specific command-line option for that setting, a user specified default configuration cache file, an environment variable, or a hard-coded default. This default value may then be accepted by entering "return", or overridden by the user. Values are checked when possible for legality and consistency.	
`-mpi_inc dir`	dir is the directory that contains the MPI library include files. Only SPMD versions of CAM require MPI. The MPI include directory can also be specified by setting the environment variable INC_MPI.	/usr/local/include except on IBM systems. The IBM Fortran compilers mpxlf90 and mpxlf90_r have the MPI include file location built in.
`-mpi_lib dir`	dir is the directory that contains the MPI library. Only SPMD versions of CAM require MPI. The MPI library directory can also be specified by setting the environment variable LIB_MPI.	/usr/local/lib except on IBM systems. The IBM Fortran compilers mpxlf90 and mpxlf90_r have the MPI library location built in.
`-nadv num`	num is the number of advected constituents. This value must be at least 3 because water vapor, cloud liquid, and cloud ice are always advected constituents.	3

Option Name	Description	Default
`-nc_inc dir`	dir is the directory that contains the NetCDF library include files. All configurations of CAM require NetCDF. The NetCDF include directory can also be specified by setting the environment variable INC_NETCDF.	/usr/local/include
`-nc_lib dir`	dir is the directory that contains the NetCDF library. All configurations of CAM require NetCDF. The NetCDF library directory can also be specified by setting the environment variable LIB_NETCDF.	/usr/local/lib
`-nlat num`	num is the number of model grid latitudes. This option is only recognized when the -res option is set to custom.	64
`-nlev num`	num is the number of model vertical layers.	26
`-nlon num`	num is the number of distinct model grid longitudes. This option is only recognized when the -res option is set to custom.	128
`-nnadv num`	num is the number of non-advected constituents. The default value is 0.	0
`-ocn name`	Select an ocean model. `name` can be either `dom`, which indicates the data ocean model, or `som`, which indicates the slab ocean model.	`dom`
`-pergro`	Configure CAM to enable perturbation growth experiments. Note that this option disables parts of the CAM code that have been found to produce rapid growth of roundoff size perturbations.	

Option Name	Description	Default
`-res name`	name specifies the horizontal resolution. For spectral dynamics the horizonal grid is Gaussian and is specified as nlat x nlon where nlat is the number of Gaussian latitudes and nlon is the number of distinct longitudes. For finite-volume dynamics the meridional grid is equally spaced and includes the pole points. It is specified as dlatxdlon where dlat is the latitude cell size and dlon is the longitude cell size, both in degrees. All of the valid resolutions are listed in the resolution_parameters.xml file in the configuration script directory. Commonly used resolutions include 48x96, 64x128, and 128x256 for the spectral dycores, and 2x2.5 for the FV dycore. To configure CAM for a resolution that is not in the resolution_parameters.xml file the value of -res must be set to `custom`. NOTE: some resolutions recognized by this option are for development purposes only. The recognition of a resolution by this option does NOT imply the existence of a validated control run.	64x128
`-s | -silent`	Turns off all output to STDOUT. Fatal error messages will still be issued to STDERR.	
`-[no]smp`	-smp enables an SMP configuration of CAM (via openMP). `-nosmp` disables an SMP configuration of CAM.	SMP is enabled by default only on IBM and SGI systems.
`-[no]spmd`	`-spmd` enables an SPMD configuration of CAM (via MPI). `-nospmd` disables an SPMD configuration of CAM.	SPMD is enabled by default only on IBM systems.
`-[no]test`	-test enables testing the Fortran compiler and external libraries and `-notest` disables testing. The tests are: 1. Check that the Fortran compiler will successfully build a "hello world" program that uses Fortran 90 module syntax. 2. Check that the Fortran compiler will successfully build a test program that contains an external reference to a NetCDF library function. 3. Check that the Fortran compiler will successfully build a test program that contains an external reference to an MPI library function, if SPMD is enabled.	testing off except in interactive prompting mode

Option Name	Description	Default
`-trk num`	num is the spectral resolution parameter that specifies the highest degree of the associated Legendre polynomials. This option is only recognized when the -res option is set to custom.	42
`-trm num`	num is the spectral resolution parameter that specifies the maximum Fourier wavenumber. This option is only recognized when the -res option is set to custom.	42
`-trn num`	num is the spectral resolution parameter that specifies the highest degree of the Legendre polynomials for m=0. This option is only recognized when the -res option is set to custom.	42
`-usr_src dir1[,dir2[...]]`	dir1[,dir2[,dir3[...]]] specifies the directories containing user source code. These directories will be placed, in the order in which they are specified, at the beginning of the Filepath file. The Filepath file is used by the CAM Makefile to determine which source files will be compiled. The list of source files is comprised of all files with .F90, .F, or .c extensions in each directory listed in Filepath plus the current directory. The Filepath file is also used by the CAM Makefile to determine which directories will be searched when looking for a source file that can be used to build an object file. The search begins in the current directory, and then proceeds to the directories in the Filepath file, in the order in which they are specified. The first file found will be the one used by make to create the object file.	none
`-v num`	num specifies the verbosity level of the output to STDOUT. Level 1 echos only the names of the files produced by configure. Level 2 adds echoing of the configuration specifications associated with a standard build of CAM. Level 3 adds echoing of the specifications associated with an "expert" build of CAM.	1
`-version`	Echo the CVS tag name that was used to check out the CAM distribution, then exit. If no tag was used for the check-out then the string *Name* will be echoed.	

Option Name	Description	Default
-x	Turns on the "expert" interactive mode. See the -i option for more details.	

2.1.2 Environment variables used by configure

The environment variables recognized by `configure` are presented in Table 2.2.

Table 2.2: **Environment variables used by configure**

Variable Name	Description
CAM_ROOT	The root directory of the CAM distribution. The directory containing the configuration scripts is $CAM_ROOT/models/atm/cam/bld.
CAM_CFGDIR	The directory that contains the CAM configuration scripts. This is provided only for the special case that the configuration scripts are taken from a directory outside of the CAM distribution.
ESMF_ROOT	Root directory to the ESMF source code.
INC_NETCDF	Directory containing the NetCDF include files.
LIB_NETCDF	Directory containing the NetCDF library.
INC_MPI	Directory containing the MPI include files. This is only required when CAM is built with SPMD enabled.
LIB_MPI	Directory containing the MPI library. This is only required when CAM is built with SPMD enabled.
USER_FC	User specified Fortran compiler. Overrides Makefile default.

Variable Name	Description
USER_CC	User specified C compiler. Overrides Makefile default. Currently only recognized on linux systems.

2.2 The `build-namelist` utility

The `build-namelist` utility builds namelists which specify run-time details for CAM and CLM. These are written to a single file (by default, the file `namelist` in the directory from which `build-namelist` is invoked).

The only required input for `build-namelist` is a configuration cache file produced by a previous invocation of `configure` (`./config_cache.xml` by default). `build-namelist` looks at this file to determine the features of the CAM executable, such as the dynamical core and horizontal resolution, that affect the default specifications for namelist variables. The default values themselves are specified in the files *DefaultCAMEXPNamelist.xml* and *DefaultCLMEXPNamelist.xml* in the CAM configuration script directory.

The user can specify namelist values that are not set by default, or can override the values that are set by default, in a number of ways. The method with highest precedence is to use the specific command-line options indicated in Table 2.3. The next highest precedence is given to any values set on the command-line using the `-namelist` option. Finally, at the lowest precedence, the default values are used. There is also an interactive prompting option (`-i`) which allows the user to view the namelist produced by the default and command-line settings, and make final changes.

2.2.1 Options to build-namelist

To get a list of all available options, type `build-namelist --help`. Available options are also provided here in Table 2.3.

All options may be specified with either one or two leading dashes, e.g., `-help` or `--help`. When multiple options are listed separated by a vertical bar (`|`), either version may be used.

Table 2.3: **Command line arguments to `build-namelist`**

Option Name	Description	Default

Option Name	Description	Default
`-cam_cfg dir`	dir is the directory that contains the CAM configuration scripts, which includes perl modules as well as various defaults files that are required by build-namelist. It is possible (but not recommended) to move build-namelist to another directory, but then the configuration script directory must be explicitly specified so that build-namelist can find its support files. Normally this directory will be determined by looking at the pathname that is used to invoke build-namelist (assuming that build-namelist has not been moved from the configuration script directory). If build-namelist is not in the configuration script directory, then that directory can be specified either by this option, by setting the environment variable CAM_CFGDIR to the configuration script directory, or by setting the environment variable CAM_ROOT to the root directory of the CAM distribution assuming that the configuration directory is $CAM_ROOT/models/atm/cam/bld.	directory part of the absolute pathname used to invoke build-namelist
`-case name`	name is the case identifier for the CAM run (up to 32 characters). This value is used to set the caseid variable in the CAM namelist.	camrun
`-config file`	file is a configuration cache file produced by the configure script. build-namelist obtains the configuration of the CAM executable from this file.	config_cache.xml
`-csmdata dir`	dir is the root directory for the default initial and boundary datasets supplied with the CAM distribution. This directory can also be specified by setting the CSMDATA environment variable. It is assumed that the root directory will contain the subdirectories atm/cam2 for CAM datasets and lnd/clm2 for CLM datasets.	/fs/cgd/csm/ inputdata. This value is set in the files DefaultCAMEXPNamelist.xml and DefaultCLMEXPNamelist.xml in the CAM configuration script directory.
`-h | -help`	Print usage to STDOUT.	

Option Name	Description	Default
-i | -interactive	Turns on interactive prompting to modify a namelist.	
-infile file	file is a namelist file to read values from. All values read from this file will appear in the output namelist unless they are overridden by other values having higher precedence.	none
-namelist namelist	namelist is a string that contains namelist settings using valid namelist syntax, e.g., -namelist "&camexp nelapse=-10, empty_htapes=.true. /" Namelist values set on the command-line take precedence over values read from a file specified with the -infile option.	none
-o file	file is the filename of the output namelist.	namelist
-runtype name	name specifies the type of simulation. Valid values are initial, restart, or branch.	initial
-s | -silent	Turns off all output to STDOUT. Fatal error messages will still be issued to STDERR.	
-test	Enable checking that initial and boundary datasets exist on local filesystem.	no checking
-v num	num specifies the verbosity level of the output to STDOUT. The default (1) echos only the name of the file produced by build-namelist. Level 2 adds echoing of the results from the -test option, and level 3 adds echoing of the default values read from the files in the CAM configuration script directory.	1

2.2.2 Environment variables used by build-namelist

The environment variables recognized by `build-namelist` are presented in Table 2.4.

Table 2.4: **Environment variables used by `build-namelist`**

Variable Name	Description
`CSMDATA`	The root directory for the default initial and boundary datasets supplied with the CAM distribution. It is assumed that the root directory will contain the subdirectories atm/cam2 for CAM datasets and lnd/clm2 for CLM datasets.
`CAM_ROOT`	The root directory of the CAM distribution. The directory containing the configuration scripts is $CAM_ROOT/models/atm/cam/bld.
`CAM_CFGDIR`	The directory that contains the CAM configuration scripts. This is provided only for the special case that the configuration scripts are taken from a directory outside of the CAM distribution.

2.3 Use Cases

This section provides a few examples of using `configure` and `build-namelist` to set up a variety of model runs. These examples were chosen to illustrate many of the configuration and namelist options described above and in the section named **CAM Namelist Variables** (B). They do not discuss the intricacies of running CAM 3.0 on its various supported platforms; a discussion of that topic may be found in the section named **Sample Run Scripts** (2.4). The following use cases have been tested on a linux PC using the Portland Group compiler (pfg90).

These examples assume a stand-alone configuration of CAM 3.0. Furthermore, **it is assumed** that the root directory for the CAM 3.0 dataset distribution is `/data`. (The root directory contains the subdirectories `atm` and `lnd`.)

The examples in this section **all assume** that the CAM configuration script directory is located in the standard place within the CAM distribution. It is also assumed that the CAM configuration directory is not in the user's `PATH` environment variable. For this reason the `configure` script **must** be invoked using an absolute pathname. The script attempts to use the pathname that it was invoked with to determine the configuration script directory. If this is successful then `configure` can determine the CAM root directory without the user needing to set the CAM_ROOT environment variable.

2.3.1 Configuring and running the default CAM executable

The following interactive C shell session builds a default production version of CAM. The shell variable `tmpdir` is set to a working directory on the user's system, and `camcfg` is set to the CAM configuration directory (`$CAM_ROOT/models/atm/cam/bld`). The following output is from a linux system, but will appear similar on other machines.

```
% cd $tmpdir
% $camcfg/configure -test
```

```
creating /big/data/temp/Filepath
creating /big/data/temp/params.h
creating /big/data/temp/misc.h
creating /big/data/temp/preproc.h
creating /big/data/temp/Makefile
creating /big/data/temp/config_cache.xml
Looking for a valid GNU make... using gmake
Testing for Fortran 90 compatible compiler... using pgf90
Testing NetCDF library... ok
configure done.
% gmake -j2  >&! make.out
% setenv CSMDATA /data
% $camcfg/build-namelist -test
Write out namelist to: namelist
% ./cam < namelist >&! output.txt
```

We started by changing into the directory in which the CAM executable will be built. All the files produced by `configure` except for the cache file are required to be in the CAM build directory, so it is generally easiest to be in that directory when `configure` is invoked. This example was carried out on a linux machine. We recommend using the `-test` option the first time CAM is built on any machine. This will check that the environment is properly set up so that the Fortran compiler works and can successfully link to the NetCDF and MPI (if SPMD enabled) libraries. The testing tells us that `gmake` is a GNU Make on this machine, that the Fortran compiler is `pgf90`, and that the compiler can successfully reference the NetCDF library.

We then issued the `gmake` command with a `-j2` option, which tells `gmake` to use 2 processors for the build. Output from the make, including the messages issued to STDERR, are redirected to the file `make.out`. In the event of an error during the build, the `make.out` file will contain the command that was issued by `gmake` that resulted in the error.

Next we set the environment variable CSMDATA to point to the root of the input data distribution (where we unpacked the intput dataset tar files).

Next we issued the `build-namelist` command. The first time a namelist for a particular CAM configuration is produced, we recommend using the `-test` option which checks whether the initial and boundary datasets exist on a local filesystem. If they do not then a warning is issued to inform the user which datasets must be copied to the directory from which CAM will be run.

The execution of `build-namelist` with `-test` produces the following default namelist:

```
&camexp
ABSEMS_DATA            = '/data/atm/cam2/rad/abs_ems_factors_fastvx.c030508.nc'
AEROPTICS              = '/data/atm/cam2/rad/AerosolOptics_c040105.nc'
BNDTVAER               = '/data/atm/cam2/rad/AerosolMass_V_64x128_clim_c031022.nc'
BNDTVDMS               = '/data/atm/cam2/rad/DMS_emissions_64x128_c030722.nc'
BNDTVGHG               = '/data/atm/cam2/ggas/ghg_1870_2100_c040122.nc'


 bndtvo         = '/data/atm/cam2/ozone/pcmdio3.r8.64x1_L60_clim_c970515.nc'
 bndtvoxid              = '/data/atm/cam2/rad/oxid_3d_64x128_L26_c030722.nc'
 bndtvs         = '/data/atm/cam2/sst/sst_HadOIBl_bc_64x128_clim_c020411.nc'
 bndtvscon              = '/data/atm/cam2/rad/scon_1870_2100_c040122.nc'
 bndtvsox               = '/data/atm/cam2/rad/SOx_emissions_64x128_L2_c030722.nc'
 bndtvvolc              = '/data/atm/cam2/rad/VolcanicMass_1870-1999_64x1_L18_c040115.nc'
```

```
 caseid         = 'camrun'
 iyear_ad               = 1950
 ncdata         = '/data/atm/cam2/inic/gaus/cami_0000-09-01_64x128_L26_c030918.nc'
 nelapse                = -1
 nsrest         = 0
/
&clmexp
 finidat                = '/data/lnd/clm2/inidata_2.1/cam/clmi_0000-09-01_64x128_T42_USGS_c030609.nc'
 fpftcon                = '/data/lnd/clm2/pftdata/pft-physiology'
 fsurdat                = '/data/lnd/clm2/srfdata/cam/clms_64x128_USGS_c030605.nc'
/
```

`build-namelist` used the configuration cache file (*config_cache.xml*) that was produced by `configure` to determine the dynamics package, land model, and resolution of the CAM executable. This information was used to choose the default initial and boundary datasets. The default run type is an initial run (`nsrest=0`), the run length is 1 day beyond the date of the initial conditions (`nelapse=-1`), and the valid year for the calculated orbital parameters is 1950 (`iyear_ad=1950`).

The absence of warnings from `build-namelist` indicates that all the initial and boundary datasets were found on the local filesystem.

Finally, we execute the `cam` executable and tell it to read its standard input from the namelist file. On systems that support running in SPMD mode this command line will be more complicated, and will typically involve a `poe` (on NCAR's IBM AIX machines) or a `mpirun` (on linux clusters) command. Please see the section named **Sample Run Scripts** (2.4) for details.

When the model is done executing, the directory should contain the files:

```
camrun.cam2.r.0000-09-02-00000
camrun.cam2.rh0.0000-09-02-00000
camrun.clm2.r.0000-09-02-00000
```

These are, respectively, a CAM master restart file, a CAM history buffer restart file, and a CLM restart file. There is no regular CAM history file because by default these are output monthly, and this was only a one day run.

2.3.2 Using the Slab Ocean Model (SOM)

The default prescribed-SST, prescribed-ice configuration of the standalone CAM model can be swapped out and replaced with a mixed-layer slab ocean model (SOM). Mixed layer temperature is the prognostic variable output from SOM. The thermodynamic ice model configured with SOM is the same as in CAM when run in prescribed-ice mode, with two additions. First, ice fraction is predicted rather than read from a boundary dataset. Second, ice thickness is also predicted, rather than fixed at 2 meters in the northern hemisphere and 1 meter in the southern hemisphere. Since SOM-configured CAM is run on the same horizontal grid as CAM, the CCSM flux coupler is not used.

Before a SOM run can be initiated, a boundary dataset must be built which contains SOM-specific information. Required steps are:

1) Define annually averaged mixed layer depths at the target horizontal resolution. Instructions for this procedure are contained in subdirectory models/atm/cam/tools/definemld of the CAM source tree. From this directory, an example sequence of commands to define mixed layer depths (after running gmake) might be:

```
./definemld1x1 -m mldfile.nc -v
./definemldbdy -s 10 -v -i mldfile.nc -o sst.nc
```

The first command creates mixed layer depth information at a horizontal resolution of 1 degree by 1 degree from ASCII input files, which are provided in the directory. The next command averages these data to the horizontal grid defined by a template file (sst.nc in the example). The mixed layer depths are written to the template file. A good choice for a template file is an existing prescribed ice/SST dataset at the desired resolution.

2) Define surface flux balance information from a control run which was done with prescribed ice and SSTs. Detailed build and execute instructions are contained in subdirectory models/atm/cam/tools/defineqflux. After building the executable, the command to construct the final boundary dataset looks something like:

```
./defineqflux -f firstfile_fromcontrolrun -l lastfile_fromcontrolrun -s sst.nc -v
```

The sequence of monthly history files from a control run is defined by the arguments to "-f" and "-l". All monthly data defined by these time boundaries must be in the working directory before running defineqflux. Output flux balance variables are written to the output file defined by the "-s" argument. Both the control run and the output file must be at the same horizontal resolution.

Example boundary dataset sst.nc is now ready for use by a SOM-enabled CAM. To configure CAM to be run in SOM mode, Filepath must be modified to include som instead of dom. Secondly, cpp token `COUP_SOM` must be defined in the file misc.h. Both of these tasks can be accomplished by running `configure` with "-ocn som".

3) Once CAM has been built with SOM as the underlying ocean model, namelist variable "bndtvs" needs to point to the SOM-specific boundary dataset described above. Since ocean surface temperature (i.e. mixed layer temperature) is a prognostic variable when SOM is enabled, model equilibration times are generally much longer than when SSTs and ice fraction are prescribed. Forty years or more of integration time may be needed for the model to reach a quasi-equilibrium state. Annually averaged mixed-layer depths (described above) are capped at 200 meters to prevent even longer equilibration times.

2.4 Sample Run Scripts

Sample run scripts are provided that illustrate how to use `configure` and `build-namelist` to set up a production run on an IBM-SP, an SGI Origin, or a PC-Linux platform.

Before we present the example run scripts, we first discuss details of the multitasking strategy employed in CAM 3.0. Both shared-memory multitasking (using OpenMP) and distributed memory (using MPI) multitasking is allowed. Hybrid-mode enables both shared-memory (for CPU's within a node) and distributed-memory (between nodes) multitasking.

Shared-memory multitasking requires the run-time specification of the number of processors to use (by setting the environment variable $OMP_NUM_THREADS). Distributed memory

multitasking involves the use of the MPI message-passing programming paradigm. The message-passing code has been validated on IBM SP3, SGI (IRIX), and Linux platforms. Different implementations of MPI handle the settings of the number of tasks in different ways.

The sample run scripts are contained in the directory $CAM_ROOT/models/atm/cam/bld/ in the files `run-ibm.csh`, `run-sgi.csh`, and `run-pc.csh`. They are set up to build and run the default CAM configuration, i.e., Eulerian dynamics at T42 spectral resolution (about 2.8 degree grid spacing) with 26 vertical levels, CLM2 land model, and CSIM4 ice model. The following list describes the features of the scripts, and how to customize them.

Root directory of CAM source distribution The shell variable `CAM_ROOT` must be set to the root directory of the CAM distribution. This variable is used to set the directory of the configuration utilities which is assumed to be `$CAM_ROOT/models/atm/cam/bld`.

Root directory of CAM dataset distribution The environment variable `CSMDATA` must be set to the root directory of the CAM dataset distribution. The value in the run scripts is appropriate for runs at NCAR on machines that mount the /fs filesystem. `CSMDATA` is used by the `build-namelist` utility to specify the locations of initial and boundary datasets used by CAM and CLM2.

CAM work directory The work directory is specified by the shell variable `wrkdir`. The defaults are appropriate for NCAR machines and are large capacity temporary filesystems. The scripts assume that the CAM build and run directories will be subdirectories of the work directory.

CAM run directory The run directory is specified by the shell variable `rundir`. By default this is set to a subdirectory of the work directory which has the same name as the run's case name (which is a required namelist input). The script will create this directory if it doesn't exist, and will exit if an error is encountered trying to create the directory.

CAM build directory The build directory is specified by the shell variable `blddir`. By default this is set to the `bld` subdirectory of the run directory. The script will create this directory if it doesn't exist, and will exit if an error is encountered trying to create the directory.

Configuration of CAM The configuration of CAM is done by calling the `configure` utility. The default configuration may be changed by supplying the appropriate arguments to `configure`.

Building CAM The script will check for an executable named `cam` in the build directory. If no executable exists then the script changes to the build directory and issues the `configure` and `gmake` commands. If the GNU make command on your system is not called "gmake", or if it is not in your PATH, then the `gmake` in the script will need to be changed to the appropriate name. `gmake` is invoked with the `-jn` option where `n` is the number of simultaneous compilation commands issued. This number should not be larger than the number of processors available for the build.

2.4.1 Use of C preprocessor tokens

`cpp` directives of the form `#include`, `#if defined`, etc., are used to enhance portability, and allow for the implementation of distinct blocks of platform-specific code within a single file. Header files, such as `misc.h`, are included with `#include` statements within the source code. When `gmake` is invoked, the C preprocessor includes or excludes blocks of code depending on which `cpp` tokens have been defined. `cpp` directives are also used to perform textual substitution for resolution-specific parameters in the code. The format of these `cpp` tokens follows standard `cpp` protocol in that they are all uppercase versions of the Fortran variables, which they define. Thus, a code statement like

```
parameter(plat = PLAT)
```

will result in the following processed line (for standard T42 resolution).

```
parameter(plat = 64)
```

2.4.2 Details of the gmake procedure

`gmake` invokes the utility `mkSrcfiles` to generate a list of source files (written to the file `Srcfiles`) using each directory listed in `Filepath`. `gmake` then invokes the utility `mkDepends` to create a dependency file (written to the file `Depends`) in the CAM build directory. If a file listed as a dependency does not exist in the CAM build directory, `gmake` searches the directories contained in `Filepath`, in the order given, for a file with that name. The first file found satisfies the dependency. If user-modified code is to be introduced, `Filepath` should contain, as the first entry (or entries), the directory containing the user code. User code directories are specified with the `-usr_src` option to `configure`.

A parallel `gmake` is achieved in the build scripts by using `gmake` with the `-j` option, which specifies the number of jobs (commands) to run simultaneously.

The output from the build is written to the file `MAKE.out` in the build directory. If a build fails this is the first place to look for information.

Rebuilding CAM The scripts are set up to *not* rebuild CAM if an executable already exists. To use the script in the situation where a rebuild of only the code (and its dependencies) that has been modified is desired, the script must be edited by commenting out the conditional statement that tests for the existence of a `cam` executable. Also, the `configure` command must be commented out since rerunning it will update the configuration files (e.g., misc.h) that all the source files depend on, and hence will result in a complete rebuild.

Creating the namelist The script will change into the build directory and issue the `build-namelist` command. The reason for the change to the build directory is that that is the location of the `config_cache.xml` file produced by `configure` which is used by `build-namelist`. The namelist is created in a file called `namelist` which is written to the run directory.

The case name, run type, and elapsed length of run are set by assigning values to the shell variables `case`, `runtype`, and `nelapse` respectively. Other changes to the namelist may be made by modifying the `-namelist` argument of the `build-namelist` command.

Before discussing the namelist variables, a brief summary is given of the types of model runs. An initial run starts the model from an initial conditions dataset. As the model executes, history datasets, restart datasets and initial condition datasets are periodically written. (see "Model Output Datasets") and "Model generated initial condition dataset files" for more details).

In addition to initial simulations, there are two types of continuation runs: restart and branch. A restart run is an exact continuation of a previous simulation from its point of termination. Namelist variables other than `NESTEP`, `NELAPSE`, and `NSREST` **should never** be modified for restart runs. A branch run, on the other hand, is a new case that uses restart data from a previous simulation to begin the integration. Since a branch run is a new case, the length of the history interval and the output history fields do not have to be the same as in the control simulation. For example, the branching option can be used to output selected fields more frequently than was the case in the control run. **Only history file namelist options should be modified for branch runs**. If the user desires to modify other options, such as physics control variables, a new initial run should be done using initial datasets generated by the control run. The user should also note that any namelist variable that can be used as a part or all of a UNIX pathname (e.g. `CASEID`,`NREVSN`,..) should only contain characters that are acceptable in UNIX path names.

The set of `CLMEXP` namelist variables that must be specified depends on whether or not a CLM 3.0 surface dataset is available at the desired model resolution. If such a dataset exists, then build-namelist will provide the minimum set `CLMEXP` of namelist variables that must be specified:

- `FPFTCON` Specifying the filepath to the plant function types dataset.
- `FSURDAT` Specifying the filepath to the CLM surface dataset.

If a CLM surface dataset is to be created at run time, then the build-namelist will attempt to provide the following CLM namelist variables:

- `FPFTCON` Specifying the filepath to the plant function types dataset.
- `MKSRF_FVEGTYP` Specifying the file to the high resolution vegetation type dataset.
- `MKSRF_FSOITEX` Specifying the file to the high resolution soil texture dataset.
- `MKSRF_FSOICOL` Specifying the file to the high resolution soil color dataset.
- `MKSRF_FLANWAT` Specifying the file to the high resolution lake water dataset.
- `MKSRF_FURBAN` Specifying the file to the high resolution urban area dataset.
- `MKSRF_FGLACIER` Specifying the file to the high resolution glacier dataset.
- `MKSRF_FLAI` Specifying the file to the high resolution leaf area index dataset.

The above datasets define the time invariant surface information, but do not provide initial condition information for the land model. If initial conditions at the specified resolution are available, they will provided by build-namelist via the namelist variable "FINIDAT".

- `FINIDAT` Specifying the filepath to initial condition dataset.

NOTE: When the user provides FSURDAT or FINIDAT datasets to the model the surface land mask and orography MUST agree with the land mask and

orography of the atmospheric model initial conditions dataset (ncdata). Furthermore, the land informaton must agree between the FINIDAT and FSURDAT datasets as well.

Running on an IBM The default parallelization on the IBM is hybrid MPI/OpenMP. The script is set up for NCAR's machine which has 4 processor nodes. One MPI process (task) is assigned to each node and by default the number of threads is set to the number of processors on each node. The number of nodes is set by either the `#@node` loadleveler variable if running in batch mode, or by the `MP_NODES` environment variable if running interactively. For the spectral dynamical cores the number of MPI processes is restricted to being a power of 2.

Changing the number of MPI processes per node is done by setting either the `#@tasks_per_node` loadleveler variable if running in batch mode, or by the `MP_TASKS_PER_NODE` environment variable if running interactively. If more than 1 MPI process is assigned to a node then the number of threads for each task should be reduced so that the number of MPI processes times the number of threads per process does not exceed the processor count on the node. The number of threads per process can be set by changing the value of the `XLSMPOPTS` environment variable. For example the number of threads is set to 2 with the string `"stack=86000000:parthds=2"`.

Running on an SGI The default parallelization on the SGI is pure OpenMP. The script is setup to use the default number of threads which is set to the number of processors on the machine, or to the number available in the queue if running in batch mode. This number can be reduced by uncommenting the setenv command for `OMP_NUM_THREADS` environment variable and setting it to the desired number. The script also uses some `module` commands which set up the system environment for NCAR machines. These commands may need to be commented out or modified on non-NCAR platforms.

Running on a PC The PC run script is set up to illustrate a pure OpenMP run. Since the default shared memory processing for the PC in the `configure` utility is none, the `-smp` option must be supplied to build CAM for an OpenMP run. The number of processes is set by the shell variable `nthreads`. The PC script does not have a batch queue submission capability.

Running batch jobs The IBM and SGI scripts both contain lines that control batch queue submission. On the IBM the loadleveler variable `#@class` must be set to a valid queue name, and the run script submitted using the `llsubmit` command. On the SGI the `-q` option to the `qsub` command must be set to a valid queue name, and the run script is submitted using the `qsub` command.

2.5 Model Input Datasets

CAM2.0.2 is a combination of atmosphere, land, ocean and sea-ice components. In what follows we discuss the input datasets required by each of these components. CSMDATA refers to the root directory where the distribution datasets have been untarred by the user.

2.5.1 Atmosphere Component Datasets

Input datasets needed for the atmospheric component provide initial state data, ozone boundary data and water vapor absorptivity/emissivity data. Only the ozone dataset contains time-variant input data which is based on a 365 day year with no leap years. All initial and boundary datasets are in NetCDF format.

In general, input dataset names follow the convention:

<type><simulation date><resolution><Spectral truncation> <Vertical levels><creation date> Where

- Simulation date = The specific date that the given dataset is valid for. In the case of initial condition datasets this is typically given in YYYY-MM-DD form. Datasets that are valid over a span of dates typically list the year range.
- Resolution = Number of latitudes by the number of longitudes for the grid being used (i.e. 64x128, 32x64 etcetera).
- Spectral truncation = For Spectral dynamical cores (eul or sld) the spectral truncation applied to the dataset (T42, T63 etcera). This is only given if the dataset in question has had spectral truncation applied to it.
- Vertical levels = Number of vertical levels (L26, L18 etc.)
- Creation date = The date the file was created in cYYMMDD format.

Initial Conditions Dataset

The initial conditions dataset is specified by namelist variable `NCDATA`. This dataset contains initial values of the prognostic variables `U`, `V`, `T`, `Q`, `PS`, `TSICE`, `SNOWHICE` and `TS1` through `TS4`, the surface geo-potential field `PHIS`, the grid-box land fraction `LANDFRAC`, the land ocean transition mask `LANDM`, and the standard deviation of geo-potential height `SGH`. When running flux-coupled the fields: TSICE, SNOWHICE, LANDFRAC and TS1 through TS4 are not included on the dataset. **Required** initial fields are outlined in 2.5.

At times it may be desirable to start the model from a more exact state of the atmosphere, with some of the fast processes (spin-up time $<$ 1 day) also represented on the initial conditions file. Table 2.6 lists **optional** fields which the model will read if they exist on the file. Fields which don't exist will be set to arbitrary values as indicated in the table:

By default, the model periodically writes an instantaneous initial conditions file containing all the fields in Table 2.5 and Table 2.6 for possible use as initial datasets in other runs. The frequency with which these datasets are written is controlled by the namelist variable `INITHIST`.

Finite-volume dynamics uses the same set of prognostic variables as shown above, except that U and V are on a staggered grid and are identified with names US and VS. When running with finite-volume dynamics, the initial dataset must contain US and VS; U and V are Ignored.

All Fields are instantaneous values. In addition to the fields listed above, the initial dataset contains information on the model date, the dimensionality of the fields, the spectral truncation, and the latitudes, longitudes, and vertical levels of the data.

A T42 (64 latitudes x 128 longitudes grid resolution) 26-level initial dataset is provided with the CAM2.0.2 distribution:

`CSMDATA/atm/cam2/inic/gaus/cami_0000-09-01_64x128_T42_L26_c020514.nc`.

Other initial datasets at different model resolutions can also be found in the directories

Table 2.5: **Atmospheric Component Initial Dataset Fields**

Field Name	**Variable Name**	**Description**
Multi-Level Fields		
T	T	Temperature (K)
U	u	Zonal Wind component (m/sec)
V	v	Meridional Wind component (m/sec)
Q	q	Water vapor specific humidity (Kg_{H2O}/Kg_{air})
Single Level Fields		
PHIS	ϕ_s	Surface geo-potential (m^2/s^2)
PS	P_s	Surface pressure (Pa)
SGH	SGH	Standard deviation of geo-potential height (m)
LANDM	landm	Land Ocean transition mask: LANDM = 0 indicates ocean, LANDM = 1 indicates land, and 0 < LANDM < 1 indicates a transition region.
TS	T_s	Surface temperature.
TSICE	T_{ice}	CSIM sea-ice model snow/ice surface temperature (not stored on CCSM flux-coupled simulations).
SNOWHICE		Snow depth over ice
LANDFRAC		Land fraction
TS1, TS2, TS3, TS4	T_s	Four CSIM sea-ice subsurface temperature levels (not stored on CCSM flux-coupled simulations) (K)

Table 2.6: **Optional Atmospheric component initial dataset fields (representing "fast" processes)**

Field Name	Variable Name	Description
Multi-Level Fields		
CLDLIQ	*q(:,:,,ixcldliq,:)*	Cloud liquid water mass mixing ratio (Kg$_{liq}$/Kg$_{air}$); set to 0. if READTRACE=.false. or field not on file
CLDICE	*q(:,:,,ixcldice,:)*	Cloud ice mass mixing ratio (Kg$_{ice}$/Kg$_{air}$); set to 0. if READTRACE=.false. or field not on file
CLOUD	cld	Cloud fraction ; set to 0. if field not on file
QCWAT	q_{cwat}	Specific humidity associated with cloud water routines ; (Kg$_{H2O}$/Kg$_{air}$); set to Q if field not on file
TCWAT	T_{cwat}	Temperature associated with cloud water routines ; (K); set to T if field not on file
LCWAT	lcwat	Total cloud water amount (liquid + ice) associated with cloud water routines ; (Kg$_{cwat}$/Kg$_{air}$); set to "CLDLIQ+CLDICE" if field not on file
Single-Level Fields		
PBLH	pblht	PBL height (m); set to 0. if field not on file
TPERT	T_{pert}	Perturbation temperature (eddies in PBL) (K); set to 0. if field not on file
QPERT	q_{pert}	Perturbation specific humidity (eddies in PBL) (KG$_{H2O}$/KG$_{air}$); set to 0. if field not on file
TSICERAD	T_{ice_rad}	Radiatively equivalent Surface T over seaice (K); set to TSICE if field not on file
TBOT	T_{bot}	Lowest model level temperature (K); set to *T(:,plev,:)* if field not on file

CSMDATA/atm/cam2/inic/gaus
and
CSMDATA/atm/cam2/inic/fv

Ozone Dataset

The ozone boundary dataset contains ozone volume mixing ratios which are constant for a given latitude. This dataset is defined on a pressure grid (unlike the hybrid grid of the model). These values are interpolated to the model vertical levels at each model grid point. The ozone dataset is in NetCDF format and contains the fields PS and OZONE. PS is a constant 1000 mb pressure field needed for interpolation. The ozone dataset is specified by namelist variable BNDTVO. Unlike the initial conditions file, CAM is capable of interpolating the ozone dataset to any vertical and or horizontal resolution so the user need not specify new ozone datasets when changing model resolution. The default ozone dataset provided with the distribution is
CSMDATA/atm/cam2/ozone/pcmdio3.r8.64x1_L60_clim_c970515.nc.

Water vapor absorptivity/emissivity dataset

This is a lookup table for water vapor absorption. It is specified by setting the namelist variable, ABSEMS_DATA. The default dataset provided with the dataset distribution can be found in
CSMDATA/atm/cam2/rad/abs_ems_factors_fastvx.052001.nc.

Aerosol Mass dataset

[Science description if any goes here]. As of CAM2.0.2.dev40, CAM requires the input aerosol dataset to be resolution-dependent. Memory and startup CPU overheads proved to be a bottleneck in the original implementation. Now horizontal interpolation to the target CAM grid and vertical integrals are done offline. Code and README describing the procedure are in subdirectory cam/tools/interpaerosols of the CAM distribution.
CSMDATA/atm/cam2/rad/abs_ems_factors_fastvx.052001.nc.

2.5.2 Ocean Component Datasets

CAM 3.0 supports running with either a data ocean model or a prognostic slab ocean model. The data ocean model simply reads and interpolates sea surface temperature (SST) data. Running the data ocean component requires a time-variant SST dataset. The SST dataset **must be at the model resolution** and is specified by the namelist variable BNDTVS.

The standard SST dataset is a climatological dataset containing 12 monthly time samples. The model can also read multi-year SST datasets. If a multi-year SST dataset is used, the namelist variable, SST_CYC must be set to .false. (it's default value is .true.).

An SST dataset on a (64 X 128) Gaussian grid can be found in
CSMDATA/atm/cam2/sst/sst_HadOIBl_bc_64x128_clim_c020411.nc.
Other SST datasets, at different model resolutions, can also be found in this directory.

The slab ocean model requires that the SST dataset specified by BNDTVS contain two additional fields specifying mixed layer depths and Q-fluxes. Preparation of these special SST datasets for the slab ocean model is described in Section 2.3.2.

2.5.3 Sea-Ice Component Datasets

The sea-ice component in CAM2.0.2 is a simplified version of the CCSM CSIM4 code. This component requires ice coverage data. The ice coverage data is contained in the same file as the SST data required by the ocean component. For the standard configuration, ice-coverage data is found in

`CSMDATA/atm/cam2/sst/sst_HadOIBl_bc_64x128_clim_c020411.nc`.

2.5.4 Land Component Datasets

For a full discussion of CLM2.1 input datasets see the CLM2.1 User's Guide at

http://www.cgd.ucar.edu/tss/clm/distribution

CLM2.1 always requires a dataset providing plant functional type physiological constants. This dataset is specified via the CLM2.1 namelist variable `FPFTCON`.

A time-invariant CLM surface dataset will be generated at run time if the CLM namelist variable `FSURDAT` is set to blank. If this is the case, additional settings must be provided for `MKSRF_FGLACIER`, `MKSRF_FLAI`, `MKSRF_FLANWAT`, `MKSRF_FSOICOL`, `MKSRF_FSOITEX`, `MKSRF_FURBAN`, `MKSRF_FVEGTYP`. If a surface dataset at the model resolution already exists, then `FSURDAT` should be set appropriately. The default land surface dataset provided with the distribution is

`CSMDATA/lnd/cam/srfdata/cam/clms_64x128_c020514.nc`.

Other provided surface datasets at other model resolutions can also be found in this directory.

CLM2.1 initial conditions will be generated at run time if the CLM2.1 namelist variable `FINIDAT` is blank (the default setting). It is important to note that due to the differences in data structures, CLM2.1 initial datasets are not in the same form as CLM2.0 initial datasets. The directory models/lnd/clm2/tools/convert_inic contains routines needed to perform the conversion of CLM2.0 initial datasets to CLM2.1 form. The README file in that directory contains necessary information to build and utilize the conversion tool. If spun up values exist for a model run, then `FINIDAT` should be correspondingly set. The default initial dataset provided with the distribution is:

`CSMDATA/lnd/clm2/inidat_2.1/cam/clmi_0000-09-01_64x128_T42_c021125.nc`.

This initial dataset **should only be used** in conjunction with the default CLM2.1 surface dataset given above.

2.6 Troubleshooting Guide

This section presents information which should help with some common problems users encounter when running the CAM.

Supported Platforms for CAM 3.0

CAM 3.0 is ported to and supported on the following platforms:

- IBM-SP
- SGI-Origin
- Linux-PC with the Portland group fortran compiler and GNU C compiler.
- Linux-PC with the Portland group fortran compiler and Portland Group C compiler.

- Linux-PC with the Lahey fortran compiler.

Known problems

- Potential problem on all platforms getting environment variables when running with distributed memory parallelism (SPMD mode).

  The model code requires that certain environment variables be available. When running in "SPMD mode" each processing task has to establish these values. Some machines have trouble when processing slaves try to get environment variables. For example, using the UNIX shell "bash" on a Linux platform with the Lahey compiler – the model fails since the slave processes are unable to obtain the environment variables. This problem can be related to the UNIX shell that the user is using. Running on Linux with the Lahey compiler and the "tcsh" shell doesn't cause the same problem that running "bash" does. Users are cautioned that other combinations of UNIX shells and platforms may result in this same problem.
- Problem compiling "history.F90" on IBM-SP with DEBUG mode and SPMD on.

  The model has undergone extensive testing on IBM platforms both at NCAR and at several other institutions. One problem discovered is a problem compiling the "history.F90" module when using the more extensive "DEBUG mode" compiler options (this mode is enabled by setting the environment variable DEBUG to "TRUE") and with the SPMD distributed memory configuration. The compiler will sometimes abort with the error "INTERNAL COMPILER ERROR". This problem is due to a problem with the IBM FORTRAN-90 compiler, and has been reported to IBM. In many cases this problem is inconsistent, so recompiling will sometimes (not always) work. As a work around the user could also compile "history.F90" without the DEBUG option on. This problem was reported to IBM and they have fixed the problem in newer versions of the compiler (or with the appropriate "E-fixes" applied to older compilers).
- Problem building ESMF library with some versions of gmake.

  Certain versions of gmake have been shown to have trouble with date-stamps. The Earth System Modeling Framework (ESMF) library has trouble with these versions and will not build. The version of gmake with the fix is 3.79.1, the bug was introduced somewhere between 3.78.1 and 3.79. So to fix the problem you merely need to update to a newer version.

General

The first step in troubleshooting a failed model run is to check the basics. Look at the logs for error messages. Make sure the model executable is up to date with any source code changes. Rebuild the model cleanly (i.e. issue a "gmake clean" before rerunning the script) if you are unsure of the state of any code. Ask yourself what has changed since the last successful run.

Other times CAM may fail for no obvious reason or perhaps the error message returned is cryptic or misleading. It has been our experience that the majority of these types of symptoms can be attributed to an incorrect allocation of hardware and/or software resources (e.g. the user sets the value of $OMP_NUM_THREADS to a value inconsistent with the number of physical CPUs per node). Most often an incorrect setting for the per-thread stack size will cause the

model to fail with a segmentation fault, allocation error, or stack pointer error. Usually the default setting for this resource is too low and must be adjusted by setting the appropriate environment variables. Values in the range of 40-70 Mbytes seem to work well on most architectures. As a simple troubleshooting step the user may try adjusting this resource, or the process stack size, for their particular application. Here is a list of suggested runtime resource settings affecting the process and/or thread stack sizes.

How to increase the stacksize on different platforms

- Compaq

```
limit stacksize unlimited
setenv MP\_STACK\_SIZE 17000000
```

- IBM

```
limit stack size unlimited
setenv XLSMPOPTS "stack=40000000"
```

- SGI origin

```
limit stack size unlimited
setenv MP\_SLAVE\_STACKSIZE 40000000
```

- SUN

```
limit stacksize unlimited
```

- PC/Linux

```
limit stacksize unlimited
setenv MPSTKZ 40000000
```

General problems on different platforms

Most distributed-memory platforms also provide runtime settings to enable a user to override the multiprocessing defaults and customize the machine parallelism to a particular application. CAM performance can be adversely affected by an incorrect configuration of the machine parallelism. The run scripts provided in the distribution create an executable that will run in a hybrid mode on distributed architectures, using MPI for communication between nodes and OpenMP directives on processes within a node. When running in hybrid mode the user should set the number of MPI tasks per node to be 1. Thread-based OpenMP multitasking will utilize all processors on the node. If the user makes the appropriate changes to the Makefile to disable OpenMP and use only MPI, the number of MPI tasks per node should be set equal to the number of physical processors per node.

At this point the model should begin compiling and executing. Appropriate log files will be generated in the `/ptmp/$LOGNAME/$CASE` directory. After a successful run of the model, the user may edit the namelist variables in run-ibm.csh to better suit their particular needs. After

successfully compiling the model, subsequent invocations of the run script will only recompile when the user makes changes to model code. The model should begin execution very quickly after gmake verifies than no code has been changed.

In addition to properly configuring machine resources, we've identified the following problems often encountered when building and running CAM on the machines here at NCAR.

- PC/Linux The number of underscore characters appended to Fortran external names is different for default configurations of the NetCDF and MPI libraries. libnetcdf.a wants to build with 1 underscore appended to Fortran external names, while libmpich.a wants to build with 2. The default CAM build procedure assumes 1 underscore. You can change this by adding the compiler directive "-Msecond_underscore" to the environment variable "USER_FLAGS". This directive produces objects with subroutine names have two underscores appended.

2.7 Running CAM as part of the CCSM coupled model system

CAM 3.0 is the atmospheric model component of the Community Climate System Model (CCSM), which also includes ocean, land, and sea-ice models, as well as a flux coupler.

Details on the use of CCSM may be found in the CCSM User's Guide.

Please note that the port of CAM to the CCSM does not use the supported configuration and control mechanisms (`configure` and `build-namelist`) described in this guide. However, portions of this User's Guide (*e.g.* **CAM Namelist Variables** (B)) may still be of interest to CCSM users.

More information on the CCSM can be found on its web page (http://www.ccsm.ucar.edu/models/ccsm3.0/ccsm/).

Chapter 3

Model Output

CAM 3.0 produces a series of NetCDF-format history files containing atmospheric gridpoint data generated during the course of a run. It also produces a series of binary restart files necessary to continue a run once it has terminated successfully and a series of initial conditions files that may be used to initialize new simulations. The formats of these datasets are described below.

3.1 Model History Files

History files contain model data values written at specified times during a run. The user can specify the frequency at which the data is written. Options are also available to record averaged, instantaneous, maximum, or minimum values on a field-by-field basis. If the user wishes to see a field written at additional time frequencies (e.g. daily, hourly), additional history files must be declared.

History files may be visualized using various commercial or freely available tools. Examples include the "CCSM Component Model Processing Suite (CMPS)", the NCAR Graphics package, FERRET, ncview, MATLAB, AVS, IDL, and Yorick. For a list of software tools for interacting with NetCDF files, view the UNIDATA NetCDF web-site:
http://my.unidata.ucar.edu/content/software/netcdf/software.html.

Aside from the default history file series the user may specify up to five additional history file series for a total of up to six history file series. The frequency at which these history file series are written as well as the contents and averaging options are specified using the same namelist variables described above for modifying the first history file series. Table 3.1 lists the fields that can be output on any of the six history files as well as which fields are output by default on the first file series.

3.1.1 Master Field List

Table 3.1 contains a list of fields, referred to as the "Master Field List", that can be written to history files.

The first column lists the names of the output variables.

The second column shows the mathematical symbol associated with the history field, as given in Collins et al. [2004].

The third column provides a brief description of the field.

In the fourth column shows the number of levels for the field. A "1" indicates a single-level field and an "N" indicates a multilevel field (on `plev` vertical levels).

The fifth column shows the default averaging flag. The flags available are: Instantaneous (I), Average (A), Maximum (X), and Minimum (M).

The sixth column in the table shows the physical units associated with each field.

The last column indicates whether the field will automatically be included on the first history file series. Fields not on by default may be included via the namelist variable `FINCL1`. Conversely, any default fields may be removed using namelist variable `FEXCL1`.

The fields are presented in alphabetical order.

Table 3.1: **Master Field List**

Field Name	**Symbol**	**Field Description**	**NL**	**AF**	**Units**	**Default**
AERASM_v		Total Aerosol Asymmetry Parameter in visible	1	I	None	
AERFWD_v		Total Aerosol Forward Scattering in visible	1	I	None	
AEROD_v		Total Aerosol Optical Depth in visible	1	I	None	
AERSSA_v		Total Aerosol Single Scattering Albedo in visible	1	I	None	
ALDIF		albedo: longwave, diffuse	1	A	1	
ALDIR		albedo: longwave, direct	1	A	1	
ASDIF		albedo: shortwave, diffuse	1	A	1	
ASDIR		albedo: shortwave, direct	1	A	1	
BGOD_v		Background Aerosol Optical Depth in visible	1	I	None	
CAROD_v		Carbon Optical Depth in visible	1	I	None	
CGH		Counter-gradient term for heat in PBL	27	A	K/m	
CGQ		Counter-gradient term for moisture in PBL	27	A	1/m	
CGS		Counter-gradient coeff on surface kinematic fluxes	27	A	s/m2	
CLDFRQ		Frequency of occurance of clouds (CLOUD ¿ 0.01)	26	A	fraction	
CLDHGH		Vertically-integrated high cloud	1	A	fraction	yes
CLDICE		Grid box averaged ice condensate amount	26	Ak	g/kg	yes
CLDLIQ		Grid box averaged liquid condensate amount	26	A	kg/kg	yes
CLDLOW		Vertically-integrated low cloud	1	A	fraction	yes
CLDMED		Vertically-integrated mid-level cloud	1	A	fraction	yes
CLDST		Stratus cloud fraction	26	A	fraction	

Field Name	Symbol	Field Description	NL	AF	Units	Default
CLDTOT		Vertically-integrated total cloud	1	A	fraction	yes
CLOUD		Cloud fraction	26	A	fraction	yes
CME		Rate of cond-evap within the cloud	26	A	kg/kg/s	
CMFDQ		Q tendency - Hack convection	26	A	kg/kg/s	yes
CMFDQR		Q tendency - shallow convection rainout	26	A	kg/kg/s	yes
CMFDT		T tendency - Hack convection	26	A	K/s	yes
CMFLQ		Moist convection total water flux	27	A	W/m2	
CMFMC		Moist convection mass flux	27	A	kg/m2/s	yes
CMFSL		Moist convection liquid water static energy flux	27	A	W/m2	
CNVCLD		Vertically integrated convective cloud amount	1	A	fraction	
CONCLD		Convective cloud cover	26	A	fraction	yes
DCCLDICE		CLDICE tendency due to moist processes	26	A	kg/kg/s	
DCCLDLIQ		CLDLIQ tendency due to moist processes	26	A	kg/kg/s	
DCQ		Q tendency due to moist processes	26	A	kg/kg/s	yes
DISED		Cloud ice tendency from sedimentation	26	A	kg/kg/s	
DLSED		Cloud liquid tendency from sedimentation	26	A	kg/kg/s	
DQP		Specific humidity tendency due to precipitation	26	A	kg/kg/s	
DQSED		Water vapor tendency from cloud sedimentation	26	A	kg/kg/s	
DTCOND		T tendency - moist processes	26	A	K/s	yes
DTH		T horizontal diffusive heating	26	A	K/s	yes
DTV		T vertical diffusion	26	A	K/s	yes
DTVKE		dT/dt vertical diffusion KE dissipation	26	A	K/s	
DUH		U horizontal diffusive heating	26	A	K/s	
DUSTOD_v		Dust Optical Depth in visible	1	I	None	
DUV		U vertical diffusion	26	A	m/s2	
DVH		V horizontal diffusive heating	26	A	K/s	
DVV		V vertical diffusion	26	A	m/s2	
EFFCLD		Effective cloud fraction	26	A	fraction	
EMIS		cloud emissivity	26	A	1	
ENGYCORR		Energy correction for over-all conservation	26	A	W/m2	

Field Name	Symbol	Field Description	NL	AF	Units	Default
ETADOT		Vertical (eta) velocity	27	A	1/s	
EVAPPCT		Percentage of Zhang-McFarlane precipitation going into evaporation	1	A	percent	
EVAPPREC		Rate of evaporation of falling precip	26	A	kg/kg/s	
EVAPSNOW		Rate of evaporation of falling snow	26	A	kg/kg/s	
FICE		Fractional ice content within cloud	26	A	fraction	yes
FLN200		Net longwave flux at 200 mb	1	A	W/m2	
FLN200C		Clearsky net longwave flux at 200 mb	1	A	W/m2	
FLNS		Net longwave flux at surface	1	A	W/m2	yes
FLNSC		Clearsky net longwave flux at surface	1	A	W/m2	yes
FLNSICE		FLNS over sea ice	1	A	W/m2	
FLNSLND		FLNS over land	1	A	W/m2	
FLNSOCN		FLNS over open ocn	1	A	W/m2	
FLNSOI		FLNS over open ocn and ice	1	A	W/m2	yes
FLNT		Net longwave flux at top of model	1	A	W/m2	yes
FLNTC		Clearsky net longwave flux at top of model	1	A	W/m2	yes
FLUT		Upwelling longwave flux at top of model	1	A	W/m2	yes
FLUTC		Clearsky upwelling longwave flux at top of model	1	A	W/m2	yes
FRACW		Relative importance of rain accreting liquid	26	A	fraction	
frc_day		Portion of time column is lit	1	I	None	
FSACI		Relative importance of snow accreting ice	26	A	fraction	
FSACW		Relative importance of snow accreting liquid	26	A	fraction	
FSAUT		Relative importance of ice autoconversion	26	A	fraction	
FSDS		Downwelling solar flux at surface	1	A	W/m2	yes
FSDSC		Clearsky downwelling solar flux at surface	1	A	W/m2	yes
FSN200		Net shortwave flux at 200 mb	1	A	W/m2	
FSN200C		Clearsky net shortwave flux at 200 mb	1	A	W/m2	
FSNIRTOA		Net near-infrared flux (Nimbus-7 WFOV) at top of atmosphere	1	A	W/m2	

Field Name	Symbol	Field Description	NL	AF	Units	Default
FSNRTOAC		Clearsky net near-infrared flux (Nimbus-7 WFOV) at top of atmosphere	1	A	W/m2	
FSNRTOAS		Net near-infrared flux (¿= 0.7 microns) at top of atmosphere	1	A	W/m2	
FSNS		Net solar flux at surface	1	A	W/m2	yes
FSNSC		Clearsky net solar flux at surface	1	A	W/m2	yes
FSNSICE		FSNS over sea ice	1	A	W/m2	
FSNSLND		FSNS over land	1	A	W/m2	
FSNSOCN		FSNS over open ocn	1	A	W/m2	
FSNSOI		FSNS over open ocn and ice	1	A	W/m2	yes
FSNT		Net solar flux at top of model	1	A	W/m2	yes
FSNTC		Clearsky net solar flux at top of model	1	A	W/m2	yes
FSNTOA		Net solar flux at top of atmosphere	1	A	W/m2	yes
FSNTOAC		Clearsky net solar flux at top of atmosphere	1	A	W/m2	yes
FU		Zonal wind forcing term	26	I	m/s	
FV		Meridional wind forcing term	26	I	m/s	
FWAUT		Relative importance of liquid autoconversion	26	A	fraction	
GCLDLWP		Grid-box cloud water path	26	A	gram/m2	yes
HEVAP		Heating from evaporation of falling precip	26	A	W/kg	
HKEIHEAT		Heating by ice and evaporation in HK convection	26	A	W/kg	
HKFLXPRC		Flux of precipitation from HK convection	27	A	kg/m2/s	
HKFLXSNW		Flux of snow from HK convection	27	A	kg/m2/s	
HKNTPRPD		Net precipitation production from HK convection	26	A	kg/kg/s	
HKNTSNPD		Net snow production from HK convection	26	A	kg/kg/s	
HMELT		Heating from snow melt	26	A	W/kg	
HPROGCLD		Heating from prognostic clouds	26	A	W/kg	
HR		Heating rate needed for d(theta)/dt computation	26	A	K/s	
HREPART		Heating from cloud ice/liquid repartitioning	26	A	W/kg	
HSED		Heating from cloud sediment evaporation	26	A	W/kg	
ICEFRAC		Fraction of sfc area covered by sea-ice	1	A	fraction	yes

Field Name	Symbol	Field Description	NL	AF	Units	Default
ICIMR		Prognostic in-cloud ice mixing ratio	26	A	kg/kg	
ICLDIWP		In-cloud ice water path	26	A	gram/m2	yes
ICLDLWP		In-cloud cloud water path (liquid and ice)	26	A	gram/m2	yes
ICWMR		Prognostic in-cloud water mixing ratio	26	A	kg/kg	
KVH		Vertical diffusion diffusivities (heat/moisture)	27	A	m2/s	
KVM		Vertical diffusion diffusivities (momentum)	27	A	m2/s	
LANDFRAC		Fraction of sfc area covered by land	1	A	fraction	yes
LANDMCOS		Land ocean transition mask: ocean (0), continent (1), transition (0-1)	1	I	unitless	
LHFLX		Surface latent heat flux	1	A	W/m2	yes
LHFLXICE		LHFLX over sea ice	1	A	W/m2	
LHFLXLND		LHFLX over land	1	A	W/m2	
LHFLXOCN		LHFLX over open ocn	1	A	W/m2	
LHFLXOI		LHFLX over open ocn and ice	1	A	W/m2	yes
LPSTEN		Surface pressure tendency	1	A	Pa/s	
LWC		Liquid Water Content	26	A	kg/m3	
LWCF		Longwave cloud forcing	1	A	W/m2	yes
LWSH		Liquid water scale height	1	A	m	
MBCPHI_V		Mass of BCPHI in and below layer	26	I	Kg/m$\hat{2}$	
MBCPHO_V		Mass of BCPHO in and below layer	26	I	Kg/m$\hat{2}$	
MBG_V		Mass of Background Aerosol in and below layer	26	I	Kg/m$\hat{2}$	
MDUST1_V		Mass of Dust bin 1 in and below layer	26	I	Kg/m$\hat{2}$	
MDUST2_V		Mass of Dust bin 2 in and below layer	26	I	Kg/m$\hat{2}$	
MDUST3_V		Mass of Dust bin 3 in and below layer	26	I	Kg/m$\hat{2}$	
MDUST4_V		Mass of Dust bin 4 in and below layer	26	I	Kg/m$\hat{2}$	
MOCPHI_V		Mass of OCPHI in and below layer	26	I	Kg/m$\hat{2}$	
MOCPHO_V		Mass of OCPHO in and below layer	26	I	Kg/m$\hat{2}$	
MQ		Water vapor mass in layer	26	A	kg/m2	

Field Name	Symbol	Field Description	NL	AF	Units	Default
MSO4		Mass concentration of SO4	26	A	gram/cm3	
MSSLT_V		Mass of Sea Salt in and below layer	26	I	Kg/m$\hat{2}$	
MSUL_V		Mass of Sulfate in and below layer	26	I	Kg/m$\hat{2}$	
MVOLC		Mass of Volcanic Aerosol in layer	26	I	Kg/m$\hat{2}$	
NSTEP		Model timestep	1	A	timestep	
O3VMR		Ozone volume mixing ratio	26	A	m3/m3	
OCNFRAC		Fraction of sfc area covered by ocean	1	A	fraction	yes
OMEGA		Vertical velocity (pressure)	26	A	Pa/s	yes
OMEGA500		Vertical velocity at 500 mbar pressure surface	1	A	Pa/s	
OMEGA850		Vertical velocity at 850 mbar pressure surface	1	A	Pa/s	
OMEGAT		Vertical heat flux	26	A	K Pa/s	yes
OMEGAU		Vertical flux of zonal momentum	26	A	K Pa/s	
PBLH		PBL height	1	A	m	yes
PBOT		Lowest model level pressure	1	A	Pa	
PCSNOW		Snow fall from prognostic clouds	1	A	m/s	
PDELDRY		Dry pressure difference between levels	26	A	Pa	yes
PHIS		Surface geopotential	1	I	m2/s2	yes
PRECC		Convective precipitation rate	1	A	m/s	yes
PRECCFRQ		Convective precipitation frequency (fraction of time where rate is ¿ 0.10mm/hr)	1	Af	raction	
PRECCINT		Convective precipitation rate (less than 0.10mm/hr is set to zero – to get intensity divide by PRECCFRQ)	1	A	mm/hr	
PRECCav		Average large-scale precipitation	1	A	m/s	
PRECL		Large-scale (stable) precipitation rate	1	A	m/s	yes
PRECLFRQ		Large-scale (stable) precipitation frequency (fraction of time where rate is ¿ 0.05mm/hr)	1	A	fraction	
PRECLINT		Large-scale (stable) precipitation rate (less than 0.05mm/hr is set to zero – to get intensity divide by PRECLFRQ)	1	A	mm/hr	
PRECLav		Average convective precipitation	1	A	m/s	

Field Name	Symbol	Field Description	NL	AF	Units	Default
PRECSC		Convective snow rate (water equivalent)	1	A	m/s	yes
PRECSED		Precipitation from cloud sedimentation	1	A	m/s	
PRECSH		Shallow Convection precipitation rate	1	A	m/s	yes
PRECSL		Large-scale (stable) snow rate (water equivalent)	1	A	m/s	yes
PRECT		Total (convective and large-scale) precipitation rate	1	A	m/s	
PRECTMX		Maximum (convective and large-scale) precipitation rate	1	Xm	/s	
PRODPREC		Rate of conversion of condensate to precip	26	A	kg/kg/s	
PS		Surface pressure	1	A	Pa	yes
PSDRY		Surface pressure	1	A	Pa	yes
PSL		Sea level pressure	1	A	Pa	yes
PS_match		Surface Pressure from aerosol climatology	1	I	N/m$\hat{2}$	
Q		Specific humidity	26	A	kg/kg	yes
Q200		Specific Humidity at 700 mbar pressure surface	1	A	kg/kg	
Q850		Specific Humidity at 850 mbar pressure surface	1	A	kg/kg	
QBOT		Lowest model level water vapor mixing ratio	1	A	kg/kg	
QC		Q tendency - shallow convection LW export	26	A	kg/kg/s	yes
QFLX		Surface water flux	1	A	kg/m2/s	yes
QPERT		Perturbation specific humidity (eddies in PBL)	1	A	kg/kg	
QRL		Longwave heating rate	26	A	K/s	yes
QRS		Solar heating rate	26	A	K/s	yes
RAINSED		Rain from cloud liquid sedimentation	1	A	m/s	
REI		effective ice particle radius	26	A	micron	
REL		effective liquid drop radius	26	A	micron	
RELHUM		Relative humidity	26	A	percent	yes
SETLWP		Prescribed liquid water path	26	A	gram/m2	
SFCLDICE		CLDICE surface flux	1	A	kg/m2/s	yes
SFCLDLIQ		CLDLIQ surface flux	1	A	kg/m2/s	yes
SFQ		Q surface flux	1	A	kg/m2/s	yes

Field Name	Symbol	Field Description	NL	AF	Units	Default
SGH		Standard deviation of orography	1	I	m	
SHFLX		Surface sensible heat flux	1	A	W/m2	yes
SHFLXICE		SHFLX over sea ice	1	A	W/m2	
SHFLXLND		SHFLX over land	1	A	W/m2	
SHFLXOCN		SHFLX over open ocn	1	A	W/m2	
SHFLXOI		SHFLX over open ocn and ice	1	A	W/m2	yes
SICTHK		Sea ice thickness	1	A	m	
SNOWHICE		Water equivalent snow depth	1	A	m	yes
SNOWHLND		Water equivalent snow depth	1	A	m	yes
SNOWSED		Snow from cloud ice sedimentation	1	A	m/s	
SOLIN		Solar insolation	1	A	W/m2	yes
SOLL		Solar downward near infrared direct to surface	1	A	W/m2	
SOLLD		Solar downward near infrared diffuse to surface	1	A	W/m2	
SOLS		Solar downward visible direct to surface	1	A	W/m2	
SOLSD		Solar downward visible diffuse to surface	1	A	W/m2	
SRFRAD		Net radiative flux at surface	1	A	W/m2	yes
SSLTOD_v		Sea Salt Optical Depth in visible	1	I	None	
SST		sea surface temperature	1	A	K	
SULFANT		Anthropogenic sulfate mass mixing ratio	26	A	kg/kg	
SULFBIO		Biogenic sulfate mass mixing ratio	26	A	kg/kg	
SULFMMR		Sulfate mass mixing ratio	26	A	kg/kg	
SULOD_v		Sulfate Optical Depth in visible	1	I	None	
SWCF		Shortwave cloud forcing	1	A	W/m2	yes
T		Temperature	26	A	K	yes
T300		Temperature at 300 mbar pressure surface	1	A	K	
T850		Temperature at 850 mbar pressure surface	1	A	K	
TAUGWX		Zonal gravity wave surface stress	1	A	N/m2	
TAUGWY		Meridional gravity wave surface stress	1	A	N/m2	
TAUTMSX		Zonal turbulent mountain surface stress	1	A	N/m2	
TAUTMSY		Meridional turbulent mountain surface stress	1	A	N/m2	

Field Name	Symbol	Field Description	NL	AF	Units	Default
TAUVIS		Total column aerosol extinction, vis band [aerosol optical depth]	1	A	unitless	
TAUX		Zonal surface stress	1	A	N/m2	yes
TAUY		Meridional surface stress	1	A	N/m2	yes
TBOT		Lowest model level temperature	1	A	K	
TEFIX		Total energy after fixer	1	A	W/m2	
TEINP		Total energy of physics input	1	A	W/m2	
TEOUT		Total energy of physics output	1	A	W/m2	
TGCLDCWP		Total grid-box cloud water path (liquid and ice)	1	A	gram/m2	
TGCLDIWP		Total grid-box cloud ice water path	1	A	gram/m2	yes
TGCLDLWP		Total grid-box cloud liquid water path	1	A	gram/m2	yes
TMQ		Total (vertically integrated) precipitatable water	1	A	kg/m2	yes
TPERT		Perturbation temperature (eddies in PBL)	1	A	K	
TREFHT		Reference height temperature	1	A	K	yes
TREFHTMN		Minimum reference height temperature over output period	1	MK		
TREFHTMX		Maximum reference height temperature over output period	1	XK		
TREFMNAV		Average of TREFHT daily minimum	1	A	K	
TREFMXAV		Average of TREFHT daily maximum	1	A	K	
TS		Surface temperature (radiative)	1	A	K	yes
TS1		TS1 subsoil temperature	1	A	K	
TS2		TS2 subsoil temperature	1	A	K	
TS3		TS3 subsoil temperature	1	A	K	
TS4		TS4 subsoil temperature	1	A	K	
TSICE		Ice temperature	1	A	K	
TSMN		Minimum surface temperature over output period	1	M	K	yes
TSMX		Maximum surface temperature over output period	1	X	K	yes
TT		Eddy temperature variance	26	A	K2	
TTEND		T tendency	26	A	K/s	
TTGWORO		T tendency - orographic gravity wave drag	26	A	K/s	
U		Zonal wind	26	A	m/s	yes

Field Name	Symbol	Field Description	NL	AF	Units	Default
U200		Zonal wind at 200 mbar pressure surface	1	A	m/s	
U850		Zonal wind at 850 mbar pressure surface	1	A	m/s	
UBOT		Lowest model level zonal wind	1	A	m/s	
US		Zonal wind, staggered	26	A	m/s	
USTAR		Surface friction velocity	1	A	m/s	
UTEND		U tendency	26	A	m/s2	
UTGWORO		U tendency - orographic gravity wave drag	26	A	m/s2	
UU		Zonal velocity squared	26	A	m2/s2	yes
V		Meridional wind	26	A	m/s	yes
V200		Meridional wind at 200 mbar pressure surface	1	A	m/s	
V850		Meridional wind at 850 mbar pressure surface	1	A	m/s	
VBOT		Lowest model level meridional wind	1	A	m/s	
VD01		Vertical diffusion of Q	26	A	kg/kg/s	yes
VDCLDICE		Vertical diffusion of CLDICE	26	A	kg/kg/s	
VDCLDLIQ		Vertical diffusion of CLDLIQ	26	A	kg/kg/s	
VOLCOD_v		Volcanic Aerosol Optical Depth in visible	1	I	None	
VQ		Meridional water transport	26	A	m/skg/kg	yes
VS		Meridional wind, staggered	26	A	m/s	
VT		Meridional heat transport	26	A	K m/s	yes
VTEND		V tendency	26	A	m/s2	
VTGWORO		V tendency - orographic gravity wave drag	26	A	m/s2	
VU		Meridional flux of zonal momentum	26	A	m2/s2	yes
VV		Meridional velocity squared	26	A	m2/s2	yes
VZ		Meridional transport of geopotential energy	26	A	m2/s	
WLWC		Weighted Liquid Water Content, prognostic (by CLDFRQ)	26	A	kg/m3	
WREL		Weighted effective radius (by CLDFRQ)	26	A	um	
WSPEED		Horizontal total wind speed	26	X	m/s	
Z050		Geopotential Z at 50 mbar pressure surface	1	A	m	
Z3		Geopotential Height (above sea level)	26	A	m	yes

Field Name	Symbol	Field Description	NL	AF	Units	Default
Z300		Geopotential Z at 300 mbar pressure surface	1	A	m	
Z500		Geopotential Z at 500 mbar pressure surface	1	A	m	
Z700		Geopotential Z at 700 mbar pressure surface	1	A	m	
ZBOT		Lowest model level height	1	A	m	
ZMDLF		Detrained liquid water from ZM convection	26	A	kg/kg/s	
ZMDQ		Q tendency - Zhang-McFarlane moist convection	26	A	kg/kg/s	
ZMDT		T tendency - Zhang-McFarlane moist convection	26	A	K/s	
ZMEIHEAT		Heating by ice and evaporation in ZM convection	26	A	W/kg	
ZMFLXPRC		Flux of precipitation from ZM convection	27	A	kg/m2/s	
ZMFLXSNW		Flux of snow from ZM convection	27	A	kg/m2/s	
ZMNTPRPD		Net precipitation production from ZM convection	26	A	kg/kg/s	
ZMNTSNPD		Net snow production from ZM convection	26	A	kg/kg/s	
ZZ		Eddy height variance	26	A	m2	

3.1.2 History Files 2 through 6

Up to six different types of history files may be written out by the model during a model run. The capability to write additional history files provides the user with the flexibility to vary the frequency at which various history data are written. Additional files may contain the same or different fields as compared with the first history file. These fields may be written on different timesteps, and have different averaging periods. Furthermore, each file may contain a different number of time samples.

3.1.3 Changing the characteristics of history files

There are several ways that namelist options can modify the characteristics of the output fields on history tapes. Output fields can be added or deleted from a file, the averaging flag can be changed and the output frequency can be varied. In addition, the number of time-samples on a file and the precision of the output data (double or single NetCDF) can be changed. To add additional fields to the first history file, the user should use the namelist variable `FINCL1`. `FEXCL1` can be used to delete fields that are on the default list of fields on the first history file series. The averaging flag may also be specified with the FINCL1 option and determines how the data is averaged over the output frequency. Values recorded for fields on a history file can be represented in one of four different ways. Data may be time averaged since the last write to

the history file, instantaneous, or appear as a point-by-point maximum or minimum over the time interval. The representation may be specified in the namelist by including a colon followed by the single character flag for each averaging type after the field name. The characters are as follows: 'A' means averaged over the interval, 'I' for instantaneous, 'M' for point-by-point minimum, and 'X' for point-by-point maximum. An example of this specification would be:

```
FINCL1 = 'T:I'
```

This specifies that temperature is to be recorded as instantaneous values on the first history file.

Other namelist variables that modify history file behavior are NHTFRQ(6) (frequency of history file writes), MFILT(6) (number of time samples per history file), and NDENS(6) (packing density). Please see Table B.1 for more information.

3.1.4 Naming the History Files

History volumes will be named according to the history filename specifier. The history filename name specifier may be specified using the namelist variable HFILENAME_SPEC (6), but by default the first history file series will contain monthly output and the filenames will be of the form

```
caseid.cam2.h0.yyyy-mm.nc
```

where caseid, yyyy, and mm correspond to the case-name, current year, and current month respectively. For example, if caseid="cambld", and current date is September, 1989 the filename becomes

```
cambld.cam2.h0.1989-09.nc
```

Non-monthly file-series are named with a full date expression as follows:

```
caseid.cam2.h#.yyyy-mm-dd-sssss.nc
```

Here, # is the file series number minus one, dd is the current day, and sssss is the number of seconds into the current day. For example, for the second file-series and a current date of September, 1, 1989, 0Z the filename becomes:

```
cambld.cam2.h1.1989-09-01-00000.nc
```

3.2 Model generated initial condition dataset files

During a model simulation, initial condition datasets are generated periodically by default. These datasets are simply history files containing instantaneous values for only those fields that are required to begin an initial run. The naming convention for these files (which is different for the other history files) is `$CASE.cam2.i.yyyy-mm-dd-sssss.nc`, where `$CASE` is the caseid, `yyyy` is the year (note, more than 4 digits will be used if needed), `mm` is the month, `dd` is the day and `sssss` is the seconds. The output frequency of the files is controlled by namelist variable `INITHIST` and is independent of the output frequency of other history files.

3.3 Restart Datasets

There are three types of restart datasets generated by the model: master, secondary, and history buffer restart files. Each dataset is in binary format and contain grid-point data and other information necessary to continue or branch a model run.

Upon restart, a simple ASCII text file (the "restart pointer file") is read to obtain the full pathname of the most recently written master restart file. Only the name of the master restart file is needed as input for a continuation run. The other files needed for restart (such as secondary restart files, or history files that need to be opened) are also listed in the restart pointer file. The Master restart file itself includes the full archive path to the files that actually need to be opened.

Master restart files are always written during a model run. A secondary restart file is written if absorptivity/emissivity is not be calculated on the first timestep after restart, and therefore must be saved on a restart dataset. For a stand-alone run this occurs if the first history file series write frequency, `NHTFRQ(1)`, is not a multiple of the absorptivity/emissivity calculation frequency, `IRADAE` (note that for a CCSM flux coupled run, only the flux coupler determines when the restart files are written). It is advisable to avoid this situation if possible, since this dataset is relatively large even for the standard T42 model.

A history buffer restart file is written in order to retain the accumulated values in the history buffers if restart files are to be written on a timestep when one or more history file time samples are not written. A separate restart dataset is written for each history file. Each history buffer restart file contains the portion of the history buffers pertaining to that history file.

All restart files have names of the form `$CASE.cam2.r.yyyy-mm-dd-sssss,` but with the ".r." changed to the appropriate restart filename. For example, ".r." is for the master restart filename, while , ".ra." corresponds to the absorptivity/emissivity restart filename, and ".rh0." is the first history file series restart filename. In the root name, `$CASE` refers to the caseid, `yyyy` corresponds to the year (note more than 4 digits for the year will be used if necessary), `mm` is the month, `dd` is the day and `sssss` is the seconds of the date `yyyymmdd`. The following is an example of the restart files written during a simulation.

- `$CASE.cam2.r.1001-01-3600` - master restart file
- `$CASE.cam2.ra.1001-01-3600` - absorptivity/emissivity restart file
- `$CASE.cam2.rh0.1001-01-3600` - history buffer restart file for first history file
- `$CASE.cam2.rh1.1001-01-3600` - history buffer restart file for second history file

3.4 Mass Store Archiving

Users running CAM on NCAR machines have the option of archiving model history files to the NCAR Mass Storage System (MSS). If history files, restart datasets, and initial conditions datasets are to be archived, they will be transferred (as a background process) to the MSS as they are completed. If namelist variable `MSS_IRT` is zero, history and restart files will not be archived. Mass Store pathnames for these transfers are generated using the `ARCHIVE_DIR` namelist setting. By default, `ARCHIVE_DIR` is set to `/$USERNAME/csm/$CASE/atm/hist`. As a result history files will be archived in the Mass Store directory

`/$USERNAME/csm/$CASE/atm/hist`

Restart files will be archived in the Mass Store directory

`/$USERNAME/csm/$CASE/atm/rest`.

And finally initial files will be archived in the Mass Store directory

/$USERNAME/csm/$CASE/atm/init.

$USERNAME is the upper-case equivalent of the user's login name, i.e., the user's root directory on the Mass Store System, and $CASE is the case identifier and is set via the namelist input. It is recommended that the user specify a non-blank write password using the namelist variable MSS_WPASS. File passwords are the only form of security available on the Mass Storage System. If the write password is not set, any other user can overwrite or change the files after they have been archived.

3.5 Model Vertical Coordinate

The vertical coordinate is a hybrid sigma-pressure system. In this system, the upper regions of the atmosphere are discretized by pressure only. Lower vertical levels use the sigma (i.e. $\mathbf{p}/\mathbf{p}_s$) vertical coordinate smoothly merged in, with the lowest levels being pure sigma. A schematic representation of the hybrid vertical coordinate and vertical indexing is presented below. Both input and output datafiles follow this format as well as internal model datastructures.

Pressure is defined as:

$\mathbf{p}_{(i,j,k)} = \mathbf{A}_k\mathbf{P}_0 + \mathbf{B}_k\mathbf{P}_s\mathbf{(i,j)}$

where $\mathbf{p}$ is the pressure at a given level and latitude, longitude grid point. The coefficients $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{P}_0$are constants. $\mathbf{P}_s$ is the model's current surface pressure. $\mathbf{P}_0$ is set in the model code. The input model initial conditions dataset sets $\mathbf{A}$ and $\mathbf{B}$ through the variables **hyam, hyai, hybm**, and **hybi.** The subscript "i" refers to interface levels, and "m" refers to the mid-point levels. "**hyam**" then refers to Hybrid level "A" coefficient on the interfaces.

More details on the theoretical nature of the vertical coordinate system can be found in Collins et al. [2004].

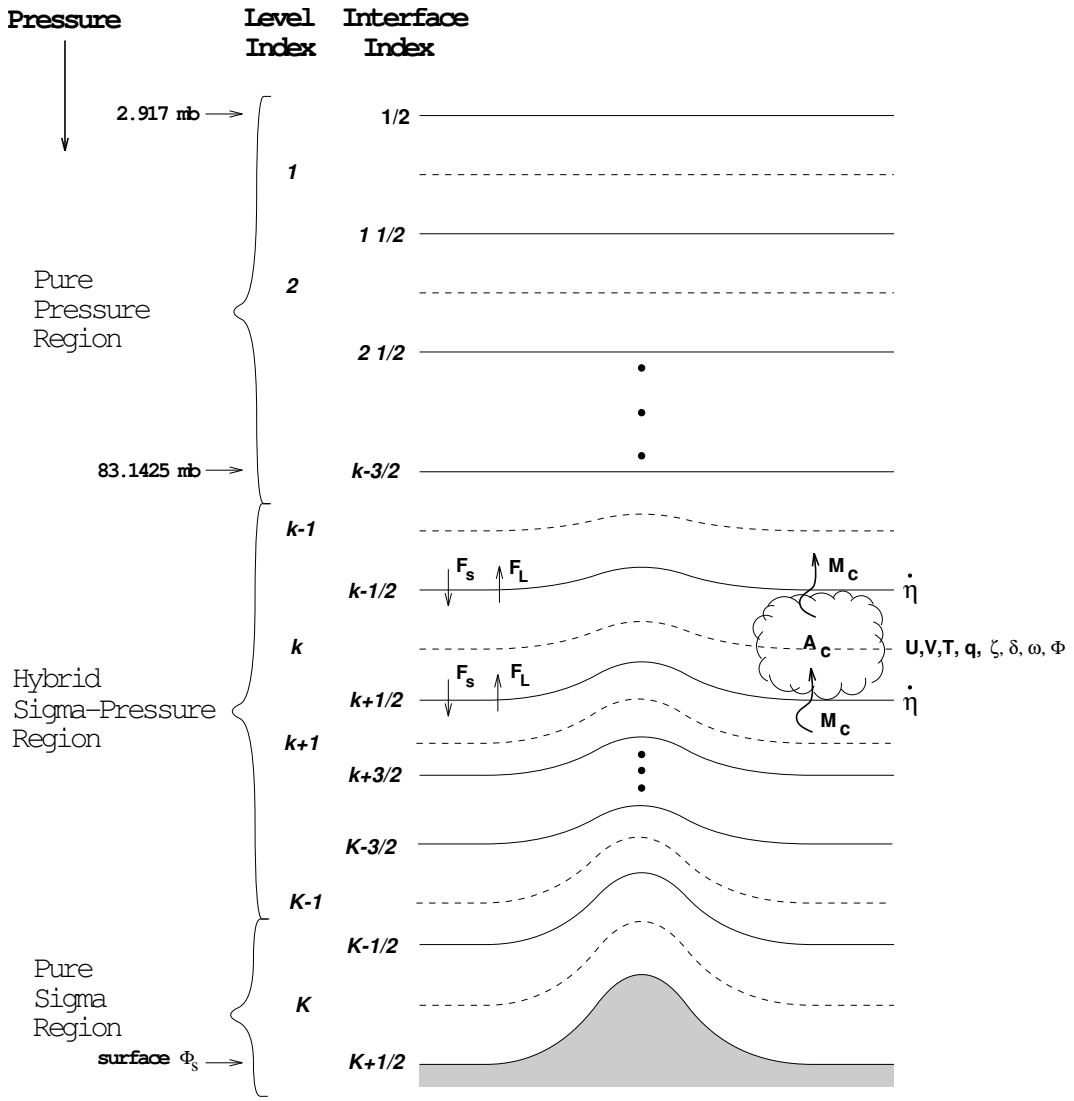

Figure 3.1: Hybrid vertical coordinate

Chapter 4

Simple Code Modifications

The most common changes to the model are the addition of new output variables, the addition of transported constituents, the modification of history file contents, the modification of the model resolution, and the addition of a new parameterization. This section provides some guidelines for making these kinds of changes. For more details on the scientific and algorithmic structure of the model see the CAM 3.0 Scientific Description.

4.1 Using the Scripts with Modified Code

If the user wishes to include modified code, a disk directory or directories should first be created where all the modified code will reside. See the "-usr_src" option in the section titled **The `configure` utility** (2.1) for documentation on configuring the model to use directories with modified source code.

4.2 Adding New Output Variables

This section describes how to add a variable to a history file. If the field is in the Master Field List (see Table 3.1), the user must ensure that there is an uncommented `outfld` call for that field and must also add it to the history output via namelist variable(s) `FINCL(1-6)`.

If the field is not on the Master Field List, the user must add it to the list by inserting an addlfd call. Six pieces of information are passed to `addfld` in an argument list:

1. Field name: 8-character field name, left-justified, alphanumeric or spaces only.
2. Field units: 8-character units description. See Table 3.1
3. Number of vertical levels in the field.
4. Default averaging flag. See Section 3.1.1 for more information.
5. Field descriptor: up to 128-characters
6. Parallel decomposition type (i.e. is this a physics or dynamics variable)

Please see the file `history.F90` for more information on the `addfld` interface. Two examples extracted from the model are shown below.

```
call addfld('TS      ', 'K       ', 1,'A', 'Surface temperature', phys\_decomp )
call addfld('U       ', 'K       ', plev,'A', 'Zonal wind'      , dyn\_decomp  )
```

Table 4.1: Units of History File Fields

Abbreviation	Description
m	meter
kg	kilogram
s	second
K	degree Kelvin
percent	percent
fraction	fraction
gram	gram
um	micro meter
N	Newton
Pa	Pascal
W	Watt

The user must then add an `outfld` call for the field at an appropriate location in the code. For example, the `outfld` call for the field `T`, taken from `diagnostics.F90`, is shown below:

```
call outfld('T       ',  state\%t, pcols, lchnk )
```

The arguments in the call to `outfld` are the 8-character field name, the variable array in which it is stored, the first dimension of the data array, and the chunk index.

Once these steps are taken, the field may be added to the desired history file by using namelist variable `FINCL(1-6)`.

4.3 Trouble-Shooting Model Changes

If the cause of abnormal termination is unclear, the user should first ensure that the model is run single-threaded with SPMD off. Abnormal termination in a multi-tasked job can result in confusing ancillary error messages.

We address several possible causes of model failure. Resource allocation errors will be addressed first, followed by remedies for suspected coding errors. Finally, analysis tools are described for physics formulation errors (i.e., where there is an error in modifications to a prognostic variable calculation).

4.3.1 Resource Allocation Errors

A system resource problem which may occur on linux architectures is that the default stack size on linux machines is sometimes too small for larger resolution runs or when running on multiple processors. The model usually fails with a segmentation fault. The user should try increasing the stack size if this problem occurs. The stack size can be set to its maximum by using the `limit` command. Typing `limit` alone will print the system resource limits. To set the stack size to its maximum type `limit stacksize unlimited` .

When running the message passing code on multiple processors it is necessary to place the `limit` command in the user's shell startup script. Since the message passing software usually starts new processes, the user must make sure that these processes have the larger stack size when started by MPI. An easy way to determine that new shells have the larger stack size is to execute the command `rsh` *`machine`* `limit` (where *machine* is the name of a computer on which to start the remote shell.)

Once the stack size has been increased try running the model again. If the stack size was too small before it should run to completion.

4.3.2 Coding Errors

We suggest that for debugging purposes only statically allocated memory locations and/or stack space be initialized to "indefinite". Furthermore,array bounds checking should be turned on if possible. The standard Makefile achieves this if `configure` is invoked with the `-debug` option.

If the model is running but producing incorrect or suspicious history files, a quick and easy-to-use diagnostic program, `cprnc`, is available. `cprnc` is available in the directory `models/atm/cam/tools`. This program provides a statistical analysis of differences in history file data. No command line arguments are required. `cprnc` compares fields of the same name on each file, printing out statistics about the number of differences found, location and magnitude of worst absolute difference, location and magnitude of worst relative difference, RMS difference, maximum and minimum field values, and average field values.

4.4 Porting the model to new resolutions

If the user needs to run the model at resolutions that aren't provided in the above set of datasets they will need additional datasets. The user will need to create an atmospheric initial-condition dataset as well as a SST dataset for this resolution. The user could do this by interpolating the datasets that are provided. Currently we do not provide tools to do this interpolation. Also when running at a different resolution land model datasets will need to be created from the high resolution datasets. These are available in the download section of the main
CAM web page (http://www.ccsm.ucar.edu/models/atm-cam).

Appendix A

Glossary

Name	Synopsis
branch run	A type of continuation run of the CAM. A branch run starts a new case using the restart files from a previous model run. This is used primarily when you wish to exactly reproduce a control simulation, but change the output fields.
CAM	Community Atmosphere Model; the latest atmosphere model code which can be run as part of the CCSM or as a stand-alone atmosphere model for climate prediction.
case	A term used to denote a model experiment, including one initial run and as many continuation runs as required to conclude the experiment. A model case is cataloged using the `CASEID` namelist variable. By default the `caseid` is included in the output filenames and mass store archive path.
cc	"C" language compiler.
CCM	Community Climate Model; the predecessor model code of CAM used in making climate predictions.
CCSM	Community Climate System Model. The set of geophysical models for atmosphere, land, ocean, and sea-ice to model the climate system. CAM is the atmosphere model component for the CCSM.
CPP	C preprocessor.
continuation run	A type of run that uses a restart file from a previous run to initialize the data fields (no initial dataset is read). Restart and branch runs are each possible types of a continuation run.
CSEG	CCSM Software Engineering Group. The group of software developers at NCAR responsible for maintaining and developing the CCSM codes.
CLM	The Community Land Model a land surface model.
CPU	Central Processing Unit.
CSIM	Community Sea-Ice Model.

Name	Synopsis
distributed memory	Multi-processing on multiple CPU's using the SPMD programming model.
DOM	Data Ocean Model. Ocean component of stand-alone CAM that reads in SST information from an input dataset.
ECMWF	European Center for Medium Range Weather Forecasting
ESMF	Earth System Modeling Framework
FV	The Finite-Volume (also referred to as Lin-Rood) dynamical core.
flux-coupled	A simulation with the atmosphere model where the model is linked to a system of geo-physical models (land, ocean, and sea-ice). When run in this manner each seperate component is built as a seperate executable, and each component communicates with each other in Multiple Program - Multiple Data (MPMD) mode. See also CCSM.
GNU	GNU's not UNIX. A set of Open-Source freely provided utilities for UNIX.
gzip	GNU decompression utility similar to UNIX uncompress. Takes a file and inflates it to it's full size so that it can be used. Files are compressed to save disk space and save on network file transfer times.
header file	A file containing CPP tokens to set the particular model resolution and configuration named with a "`.h`" suffix (e.g. `misc.h`).
heap	Memory that is dynamically allocated by the system. Unlike stack memory, heap memory can be allocated or de-allocated at any point during program execution.
history file	The output NetCDF dataset that the model produces to record the model simulation field values with time.
initial run	A startup simulation using a initial-condition dataset.
LSM	The NCAR Land Surface Model. This model was used with the CCM the predecessor to CAM.
module	A FORTRAN-90 construct containing data and subroutines and functions that operate on that data. This is a somewhat object-oriented approach to datastructures. The data can be specified as private to the module or public for access by other subroutines. Outside subroutines can specify the limited set of variables they wish to capture. This ability provides a better approach to handling data and is the preferred method for dealing with data over COMMON blocks.
MPI	Message Passing Interface. A standardized library for distributed memory parallel processing.
MPMD	Multiple Program Multiple Data. Parallel programming model with several distinct executable programs operating on different sets of data.

Name	Synopsis
mpxlf90	FORTRAN-90 compiler for IBM AIX with message passing (MPI) libraries included.
mpxlf90_r	Thread-safe FORTRAN-90 compiler for IBM AIX with message passing (MPI) libraries included, and allowing for OpenMP threading.
MSS	The NCAR Mass Store System.
multi-tasked	A program configured to execute on several distributed processors simultaneously. See distributed memory.
NCAR	National Center for Atmospheric Research
NetCDF	Network Common Data Format. Self describing, platform independent binary data format (created by UNIDATA).
OpenMP	Open specification for Multi-Processing. Set of compiler directives for shared-memory parallel processing, that is supported by most compiler venders.
PBL	Planetary Boundary Layer.
pcnst	Number of advected constituents carried in the model.
pgf90	Portland Group's FORTRAN-90 compiler,
plat	Number of Gaussian latitudes on the transform grid.
plev	Number of vertical model levels.
plon	Number of longitudes on the transform grid
restart run	A type of continuation run of the CAM. A restart run continues a previous run from its point of termination, by reading most recent restart
single-threaded	Parallel-processing term. Refers to the parts of shared memory processed code that executes on only one processor.
shared-memory	Parallel-processing term. Refers to using a machine where multiple CPU's share the same memory.
SLT	Semi-Lagrangian Transport.
SPMD	Single Program Multiple Data.
stack	Memory local to a subroutine or function.
stand-alone	Running the model not in flux-coupled mode as described above, but as a single-executable with land, thermodynamic sea-ice, and data ocean models as subroutines of the main executable.
STDERR	Standard error. Output stream that error messages are sent to. Usually, this is the users terminal.
STDOUT	Standard output. Default output stream that messages are sent to. Usually, this is the users terminal.
transform grid	A grid used to evaluate all nonlinear and diabatic forcing terms in physical space.
UNIDATA	University Atmospheric Data access project funded by the National Science Foundation (NSF).
UNIX	A modern operating system shared and supported by most supercomputers.

Appendix B

CAM Namelist Variables

A CAM model run is controlled using the CAM `build-namelist` facility described in Section 2.2. The focus of this appendix is to provide a reference for the variables that may be set through the use of `build-namelist`'s `-namelist` option.

Please see Table B.1 in the section named **Complete list of CAM namelist variables** (B.13) for a list of all valid namelist variables in alphabetical order. We also present a set of convenient subject-oriented lists broken into the following categories:

- **CAMEXP run type namelist variables** (B.1)
- **CAMEXP time management namelist variables** (B.2)
- **CAMEXP input dataset namelist variables** (B.3)
- **CAMEXP history file namelist variables** (B.4)
- **CAMEXP mass store control namelist variables** (B.5)
- **CAMEXP restart file namelist variables** (B.6)
- **CAMEXP dynamics namelist variables** (B.7)
- **CAMEXP physics namelist variables** (B.8)
- **CAMEXP non-water constituent namelist variables** (B.9)
- **CAMEXP memory namelist variables** (B.10)
- **CAMEXP performance tuning namelist variables** (B.11)
- **CAMEXP orbital parameter namelist variables** (B.12)

When using the pdf or html versions of this document, you may click on any variable name to get a complete description.

B.1 CAMEXP run type namelist variables

CASEID, CTITLE, NSREST, PERPETUAL_RUN, PERPETUAL_YMD

B.2 CAMEXP time management namelist variables

CALENDAR, DTIME, NELAPSE, NESTEP, REF_TOD, REF_YMD, START_TOD, START_YMD, STOP_TOD, STOP_YMD,

B.3 CAMEXP input dataset namelist variables

ABSEMS_DATA, AEROPTICS, BNDTVAER, BNDTVCARBONSCALE, BNDTVDMS, BNDTVGHG, BNDTVG, BNDTVOXID, BNDTVO, BNDTVSCON, BNDTVSF6, BNDTVSOX, BNDTVS, BNDTVVOLC, CO_EMIS, ISCCPDATA, NCDATA, OZNCYC, SOIL_EROD, SSTCYC

B.4 CAMEXP history file namelist variables

AVGFLAG_PERTAPE (6), EMPTY_HTAPES, FEXCL1, FEXCL2, FEXCL3, FEXCL4, FEXCL5, FEXCL6, FINCL1, FINCL2, FINCL3, FINCL4, FINCL5, FINCL6, FINCL1LONLAT, FINCL2LONLAT, FINCL3LONLAT, FINCL4LONLAT, FINCL5LONLAT, FWRTPR1, FWRTPR2, FWRTPR3, FWRTPR4, FWRTPR5, FWRTPR6, HFILENAME_SPEC (6), INITHIST, MFILT(6), NDENS(6), NHTFRQ(6), PRECC_THRESH, PRECL_THRESH

B.5 CAMEXP mass store control namelist variables

ARCHIVE_DIR, MSS_IRT, MSS_WPASS

B.6 CAMEXP restart file namelist variables

BRNCH_RETAIN_CASENAME, NREFRQ, NREVSN, REST_PFILE

B.7 CAMEXP dynamics namelist variables

DIF2, DIF4, DIVDAMPN, EPS, IORD, JORD, KMXHDC, KORD, NSPLIT, USE_ETA,

B.8 CAMEXP physics namelist variables

ADIABATIC, AERO_CARBON, AERO_FEEDBACK_CARBON, AERO_FEEDBACK_SEA_SALT, AERO_SEA_SALT, AQUA_PLANET, BGSCL_RF, CARSCL_RF, CARSCL, CH4VMR, CO2VMR, DOISCCP, DUSTSCL_RF, DUSTSCL, F11VMR, F12VMR, FLXAVE, ICE_CONSCHK_FRQ, IDEAL_PHYS, INDIRECT, IRADAE, IRADLW, IRADSW, ITSST, N2OVMR, NLVDRY, PERTLIM, PRESCRIBED_SULFUR, PROGNOSTIC_ICESNOW, PROGNOSTIC_SULFUR, RADFORCE, RAMPYEAR_GHG, RAMPYEAR_PRESCRIBED_SULFUR, RAMPYEAR_PROGNOSTIC_SULFUR, RAMPYEAR_SCON, RAMP_CO2_ANNUAL_RATE, RAMP_CO2_CAP, RAMP_CO2_START_YMD, RESET_CSIM_ICE_PROPS, SCENARIO_CARBON_SCALE, SCENARIO_GHG, SCENARIO_PRESCRIBED_SULFUR, SCENARIO_PROGNOSTIC_SULFUR, SCENARIO_SCON, SCON, SOM_CONSCHK_FRQ, SSLTSCL_RF, SSLTSCL, STRAT_VOLCANIC, SULSCL_RF, SULSCL, TAUBACK, TAUVIS, TRACE_GAS, VOLCSCL_RF, VOLCSCL

B.9 CAMEXP non-water constituent namelist variables

NUSR_ADV, NUSR_NAD, READTRACE

B.10 CAMEXP memory namelist variables

FHSTPR1, FHSTPR2, FHSTPR3, FHSTPR4, FHSTPR5, FHSTPR6, LINEBUF, NHSTPR(6), PRINT_STEP_COST

B.11 CAMEXP performance tuning namelist variables

DYN_ALLGATHER, DYN_ALLTOALL, FORCE_2D, GEOPKTRANS, MODCOMM_GEOPK, MODCOMM_TRANSPOSE, NPR_YZ, OMPNEST, PHYS_ALLTOALL, PHYS_CHNK_PER_THD, PHYS_LOADBALANCE, SWAP_COMM_ORDER, SWAP_COMM_PROTOCOL, TRACERTRANS

B.12 CAMEXP orbital parameter namelist variables

ECCEN, IYEAR_AD, MVELP, OBLIQ

B.13 Complete list of CAM namelist variables

Table B.1 displays all supported variables. Most of the variables in CAMEXP are single-valued, but some are array-valued. Included in the table are the following pieces of information:

- Variable name.
- Variable type (Char, Integer, Real, or Logical). If the variable is of type Char, then its declared length is displayed along with the variable type (*e.g.* CTITLE is 80 characters in length, therefore Char*80 is entered in the "Type" column).
- Default value.
- References to examples in Section 2.3 demonstrating usage.
- Variable description

Table B.1: **CAMEXP namelist variables**

Name	Type	Default	Ex.	Description
ABSEMS_DATA	Char*256	See Desc.		Filename of absorption/emission dataset. This file is a required input dataset. It consists of terms used for determining the absorptivity and emissivity of water vapor in the longwave parameterization of radiation. Default is set in DefaultCAMEXPNamelist.xml.
ADIABATIC	Logical	.FALSE.		If true, do not run model physics, only run the dynamical core. In this mode, water vapor is advected as a passive tracer. **Only one of ADIABATIC, IDEAL_PHYS, or AQUA_PLANET can be true.**
AEROPTICS	Char*256	See Desc.		Full pathname of dataset for time-invariant aerosol optical properties. Default is set in DefaultCAMEXPNamelist.xml.

Name	Type	Default	Ex.	Description
AERO_CARBON	Logical	.FALSE.		If true, turn on carbon prognostic aerosols.
AERO_FEEDBACK_CARBON	Logical	.FALSE.		If true, turn on feedback of carbon prognostic aerosols. May only be true if AERO_CARBON is also true.
AERO_FEEDBACK_SEA_SALT	Logical	.FALSE.		If true, turn on feedback of sea salt prognostic aerosols. May only be true if AERO_SEA_SALT is also true.
AERO_SEA_SALT	Logical	.FALSE.		If true, turn on sea salt prognostic aerosols.
AQUA_PLANET	Logical	false		Run model in "AQUA_PLANET" mode. **Only one of ADIABATIC, IDEAL_PHYS, or AQUA_PLANET can be true.**
ARCHIVE_DIR	Char*256	See Desc.		Head mass store archival directory name. Default is /$USER/csm/caseid/ (where caseid is the caseid from the namelist and $USER is the username converted to uppercase.) CAM history files will be stored under $ARCHIVE_DIR/atm/hist, restart files under $ARCHIVE_DIR/atm/rest, and initial files under $ARCHIVE_DIR/atm/init. Land-model output files will be stored under similar directory names with "lnd" to specify the land-model subdirectory.
AVGFLAG_PERTAPE (6)	Char*1	See Desc.		Sets the averaging flag for all variables on a particular history file series. Default is to use default averaging flags for each variable Valid values are A, I, X, and M, indicating Average (A), Instantaneous (I), Maximum (X), and Minimum (M).
BGSCL_RF	Real	0		Set background aerosol scaling factor for radiative forcing calculation. This does not affect mmr's used for the climate integration, and are is used when RADFORCE is true.
BNDTVAER	Char*256	See Desc.		Full pathname of time-variant boundary dataset for aerosol mass mixing ratios. Default is set in DefaultCAMEXPNamelist.xml.
BNDTVCARBONSCALE	Char*256	See Desc.		Full pathname of time-variant boundary dataset for carbon scaling. Default is set in DefaultCAMEXPNamelist.xml.
BNDTVDMS	Char*256	See Desc.		Full pathname of time-variant boundary dataset for DMS surface emissions. This is used as an input for prognostic sulfate computations. Default is set in DefaultCAMEXPNamelist.xml.

Name	Type	Default	Ex.	Description
BNDTVGHG	Char*256	See Desc.		Full pathname of time-variant boundary dataset for greenhouse gas surface values. Default is set in DefaultCAMEXPNamelist.xml.
BNDTVG	Char*256	See Desc.		Full pathname of time-variant boundary dataset for greenhouse loss rates. (required if trace_gas is set to true). Default is set in DefaultCAMEXPNamelist.xml.
BNDTVOXID	Char*256	See Desc.		Full pathname of time-variant boundary dataset for oxidants. Default is set in DefaultCAMEXPNamelist.xml.
BNDTVO	Char*256	See Desc.		Full pathname of time-variant ozone mixing ratios boundary dataset (NetCDF format). Default is set in DefaultCAMEXPNamelist.xml.
BNDTVSCON	Char*256	See Desc.		Full pathname of time-variant boundary dataset for solar constant. Default is set in DefaultCAMEXPNamelist.xml.
BNDTVSF6	Char*256	See Desc.		Full pathname of time-variant boundary dataset for tracer test emissions (required if tracers_flag is set to true). Default is set in DefaultCAMEXPNamelist.xml.
BNDTVSOX	Char*256	See Desc.		Full pathname of time-variant boundary dataset for SOx surface emissions. Default is set in DefaultCAMEXPNamelist.xml.
BNDTVS	Char*256	See Desc.	??, ??, ??	Full pathname of time-variant sea-surface temperature and sea-ice concentration boundary dataset (NetCDF format). Default is set in DefaultCAMEXPNamelist.xml.
BNDTVVOLC	Char*256	See Desc.		Full pathname of time-variant boundary dataset for stratospheric volcanic aerosol mass mixing ratios. Default is set in DefaultCAMEXPNamelist.xml.
BRNCH_RETAIN_CASENAME	logical	false		If TRUE, use the pre-existing case name for a branch run.
CALENDAR	Char*32	NO_LEAP		Calendar type 'NO_LEAP' for consistent 365-days per year or 'GREGORIAN' to include leap-years. Note that if "GREGORIAN" is selected although leap-years will be used in the calender manager the calculation of the earth's orbit still assumes 365 day years., Valid values: "NO_LEAP" or "GREGORIAN"
CARSCL_RF	Real	0		Set carbon aerosol scaling factor for radiative forcing calculation. This does not affect mmr's used for the climate integration, and is only used when RADFORCE is true.
CARSCL	Real	1.0		Set carbon aerosol scaling factor for radiative transfer calculations. This may be used to scale the specified or prognostic aerosol mass mixing ratios.

Name	Type	Default	Ex.	Description
CASEID	Char*32	camrun	??	Case identifier (saved in history file header record) and normally used as part of the MSS path name (see the ARCHIVE_DIR namelist option). By default also used in output filenames (see the HFILENAME_SPEC namelist option). **Setting is Required.**
CH4VMR	Real	1.714e-6	??	CH_4 volume mixing ratio.
CO2VMR	Real	3.550e-4	??	CO_2 volume mixing ratio
CO_EMIS	Char*256	See Desc.		Full pathname of time-variant boundary dataset for fossil fuel carbon surface emissions. Default is set in DefaultCAMEXPNamelist.xml.
CTITLE	Char*80	None	??	Case title
DIF2	Real	See Desc.		del^2 horizontal diffusion coefficient. Default is resolution dependent, e.g. 2.5e5 for T42 EUL. Not used for fv dynamics.
DIF4	Real	See Desc.		del^4 horizontal diffusion coefficient. Default is resolution dependent, e.g. 1.0e16 for T42 EUL. Not used for fv dynamics.
DIVDAMPN	Real	0.0		Number of days (from timestep 0) to run divergence damper. Use only if model becomes dynamicallly unstable during initialization. Suggested value: 2. (Value must be >= 0.) Not used for fv dynamics.
DOISCCP	Logical	false		If true, the ISCCP cloud simulator will be run, and output added to the monthly history files.
DTIME	Real	See Desc.	??	Length of model timestep in seconds. CAUTION: Changing this variable directly impacts the physical parameterizations in the model and may impact the climate. Changing resolution usually requires a change in DTIME. Defaults are resolution-dependent, e.g 1200. for T42 EUL; 1800 for FV 2x2.5.
DUSTSCL_RF	Real	0		Set dust aerosol scaling factor for radiative forcing calculation. This does not affect mmr's used for the climate integration, and is only used when RADFORCE is true.
DUSTSCL	Real	1.0		Set dust aerosol scaling factor for radiative transfer calculations. This may be used to scale the specified or prognostic aerosol mass mixing ratios.

Name	Type	Default	Ex.	Description
DYN_ALLGATHER	Integer	0		dynamics gather option. (only used with EUL and SLD dycores).
DYN_ALLTOALL	Integer	0		dynamics transpose option. (only used with EUL and SLD dycores).
ECCEN	Real	None	??	Earth's eccentricity of orbit. (unitless: typically 0. to 0.1). **Setting is Required if IYEAR_AD not set. Not used when running as part of CCSM.**
EMPTY_HTAPES	Logical	.FALSE.		If set don't put any of the variables on the history tapes by default. Only output the variables that the user explicitly lists in the fincl# namelist items.
EPS	Real	0.06		Time filter coefficient.
F11VMR	Real	0.280e-9	??	CFC_{11} volume mixing ratio.
F12VMR	Real	0.503e-9	??	CFC_{12} volume mixing ratio.
FEXCL1, FEXCL2, FEXCL3, FEXCL4, FEXCL5, FEXCL6 (200)	Char*8	None	??	List of fields to exclude on the default history files (must be in Master Field List). The default history files include a monthly series and a daily series. For a list of what fields are included on the first history file series by default see Table 3.1.
FHSTPR1, FHSTPR2, FHSTPR3, FHSTPR4, FHSTPR5, FHSTPR6 (200)	Char*8	None		Specific fields for each history file for which to use the non-default history buffer precision.
FINCL1, FINCL2, FINCL3, FINCL4, FINCL5, FINCL6 (200)	Char*8	None	??, ??	List of fields to include on the history files. The added fields must be in Master Field List see Table 3.1 for the list of fields that can be added as well as which fields are on the first history file series by default (FINCL1). Using a ":" following a field gives the averaging flag for the output field. Valid flags are: I for instantaneous, A for average, M for minimum, and X for maximum. See Section 2.5.1.1 for more information on the averaging flag.

Name	Type	Default	Ex.	Description
FINCL1LONLAT, FINCL2LONLAT, FINCL3LONLAT, FINCL4LONLAT, FINCL5LONLAT (200)	Char*8	None		List of columns or contiguous columns at which the fincl1 fields will be output. Individual columns are specified as a string using a longitude degree (greater or equal to 0.) followed by a single character (e)ast/(w)est identifer, an underscore '_' , and a latitude degree followed by a single character (n)orth/(s)outh identifier. For example, '10e_20n' would pick the model column closest to 10 degrees east longitude by 20 degrees north latitude. A group of contiguous columns can be specified using bounding latitudes and longitudes separated by a colon. For example, '10e:20e_15n:20n' would select the model columns which fall with in the longitude range from 10 east to 20 east and the latitude range from 15 north to 20 north.
FLXAVE	Logical	See Desc.		This namelist variable is only used when running through the flux coupler. In that case, the default value is true. If true, data is sent to the flux coupler only on radiation time steps.
FORCE_2D	Integer	0		option to force transpose computation for 1D decomp. (used with Finite-Volume dynamical core only).
FWRTPR1, FWRTPR2, FWRTPR3, FWRTPR4, FWRTPR5, FWRTPR6 (200)	Char*8	None		Specific fields for each history file which will be written using the non-default precision.
GEOPKTRANS	Integer	0		Geopotential method (used with Finite-Volume dynamical core only).

Name	Type	Default	Ex.	Description
HFILENAME_SPEC (6)	Char*256	See Desc.		Array of history filename specifiers. Defaults are "%c.cam2.h%t.%y-%m", "%c.cam2.h%t.%y-%m-%d-%s", "%c.cam2.h%t.%y-%m-%d-%s", ... Filename, specifiers give generic formats for the filenames with the specific dates, file-series number, and caseid, filled in when the files are created. The following strings are expanded when the filename is created: • %c = caseid • %t = tape series number (0-5) • %y = year (normally 4 digits, more digits if needed) • %m = month • %d = day • %s = seconds into current day • %% = % symbol For example, for a simulation with caseid="test" and current date of 0000/12/31 0:00UT, the monthly average files with a filename specifier of "%c.cam2.h%t.%y-%m.nc" expand into "test.cam2.h0.0000-12.nc", daily files with a filename specifier of "%c.cam2.h%t.%y-%m-%d-%s.nc" expand to ""test.cam2.h0.0000-12-31-00000.nc". Spaces are not allowed in filename specifiers. Although the character "/" is allowed in the specifier, it will be interpreted as a directory name and the corresponding directories will have to be created in the model execution directory (directory given to configure with -cam_exedir option) before model execution.
ICE_CONSCHK_FRQ	Integer	0		This is not used only when running as part of CCSM. If n > 0, sea ice global energy checking will be done every n timesteps. If n < 0, sea ice global energy checking will be done every n days.
IDEAL_PHYS	Logical	false		If true, Run ONLY the "idealized" dynamical core of the model (dynamics + Held&Suarez-specified physics). **Only one of ADIABATIC, IDEAL_PHYS, or AQUA_PLANET can be true.**
INDIRECT	Logical	false		If true, include the indirect radiative effects of sulfate aerosols.

Name	Type	Default	Ex.	Description
INITHIST	Char*8	YEARLY		Frequency that initial files will be output: 6-hourly, daily, monthly, yearly, or never. Valid values: 'NONE', '6-HOURLY', 'DAILY', 'MONTHLY', 'YEARLY'
IORD	Integer	4		East-west transport scheme (used with Finite-Volume dynamical core only).
IRADAE	Integer	-12		Frequency of absorptivity/emissivity calculations in time steps (if positive) or model hours (if negative). To avoid having the abs/ems values saved on the restart output, this variable should divide evenly into MFILT(1)*NHTFRQ(1).
IRADLW	Integer	-1		Frequency of long-wave radiation calculation in timesteps (if positive) or model hours (if negative).
IRADSW	Integer	-1		Frequency of short-wave radiation calculation in timesteps (if positive) or model hours (if negative).
ISCCPDATA	Char*256	See Desc.		Full pathname of time-invariant dataset used by the ISCCP cloud simulator. Default is set in DefaultCAMEXPNamelist.xml.
ITSST	Integer	1		Frequency of SST update in timesteps.
IYEAR_AD	Integer	1950	**??**, **??**	Year (AD) used to compute earth's orbital parameters. If not set, then use the values from the eccen, mvelp, and obliq namelist parameters. *If only IYEAR_AD is set, orbital parameters will be computed automatically (based on Berger, 1977). If one of OBLIQ, ECCEN, OR MVELP is set, all three must be set. If all four of the above are set by the user, IYEAR_AD takes precedence.* **Setting is Required unless ECCEN, OBLIQ and MVELP are set. Not used when running as part of CCSM.**
JORD	Integer	4		North-south transport scheme (used with Finite-Volume dynamical core only).
KMXHDC	Integer	5		Number of levels over which to apply Courant limiter, starting at top of model.
KORD	Integer	4		Vertical mapping (used with Finite-Volume dynamical core only).

Name	Type	Default	Ex.	Description
LINEBUF	Logical	.FALSE.	??	If true, force buffer flush of stdout with each new-line generated (useful for debugging).
MFILT(6)	Integer	30, 30, 30, ...	??, ??, ??	Array of number of time samples to write to each history files series (a time sample is the history output from a given timestep).
MODCOMM_GEOPK	Integer	0		mod_comm geopk method (varies with mpi/mpi2 choice) (used with Finite-Volume dynamical core only).
MODCOMM_TRANSPOSE	Integer	0		mod_comm transpose method (varies with mpi/mpi2 choice) (used with Finite-Volume dynamical core only).
MSS_IRT	Integer	365	??, ??, ??, ??	Retention period in days for history and restart files stored on NCAR Mass Store. If MSS_IRT=0, no history or restart files will be archived.
MSS_WPASS	Char*8	None	??, ??	NCAR Mass Store write password for all output datasets. Must remain the same throughout a case run, or a continuation run will not update original history files.
MVELP	Real	None	??	Earth's moving vernal equinox at perihelion (degrees: 0. to 360.0). **Setting is Required if IYEAR_AD not set. Not used when running as part of CCSM.**
N2OVMR	Real	0.311e-6	??	N_2O volume mixing ratio.
NCDATA	Char*256	See Desc.	??, ??	Full pathname of initial atmospheric state dataset (NetCDF format). See Section 2.5.1 for details. Default is set in DefaultCAMEXPNamelist.xml.
NDENS(6)	Integer	2, 2, 2, ...	??, ??, ??	Array specifying output format for each history file series. Valid values are 1 or 2. "1" implies output to be written out in 64-bit NetCDF format, and "2" writes data out in 32-bit NetCDF format.
NELAPSE	Integer	None		Elapsed time to run. In time steps (positive) or days (negative) to run. May be entered instead of NESTEP or STOP_YMD and STOP_TOD. Note, NESTEP is not required or used when running as part of CCSM. **Setting is Required if NESTEP or STOP_YMD not set.**

Name	Type	Default	Ex.	Description
NESTEP	Integer	None		Ending timestep. If positive the ending absolute timestep (from the start of the initial run). If negative the ending absolute number of days to process (from the start of the initial run). Either NELAPSE, NESTEP or (STOP_YMD,STOP_TOD) must be set if not running through flux coupler. (STOP_YMD,STOP_TOD) take precedence if set. Note, NESTEP is not required or used when running as part of CCSM. **Setting is Required if NELAPSE and STOP_YMD not set.**
NHSTPR(6)	Real	8, 8, 8, ...		Default history buffer precision (8 or 4 bytes) for each history file.
NHTFRQ(6)	Integer	0, -24, -24, -24, ...	??, ??, ??	Array of write frequencies for each history files series. If NHTFRQ(1)=0, the file will be a monthly average. Only the first file series may be a monthly average. If NHTFRQ(i)>0, frequency is input as number of timesteps. If NHTFRQ(i)<0, frequency is input as number of hours. Note that NHTRFQ(1) also controls the frequency of restart dataset writes if NREFRQ = 1.
NLVDRY	Integer	3		Number of layers from the top of the model over which to do dry convective adjustment. Must be less than plev (the number of vertical levels).
NPR_YZ	Integer	0, 0, 0, 0		YZ and XY decompositions (used with Finite-Volume dynamical core only).
NREFRQ	Integer	1		Controls whether restart datasets are written. **This variable can only be 1 or 0**. If 1, restart files are written and archived for every archive of the first history file series. If this variable is 0, then no restarts are written.
NREVSN	Char*256	None		Full pathname of master restart file from which to branch. **Setting is Required for branch run.**
NSPLIT	Integer	0		Lagrangian time splits when using Finite-Volume dynamics. If zero, a best-estimate will be automatically calculated.
NSREST	Integer	0	??, ??, ??	Run type: 0=initial , 1=restart , 3=branch
NUSR_ADV	Integer	0		number of user defined advected tracers

Name	Type	Default	Ex.	Description
NUSR_NAD	Integer	0		number of user defined non-advected tracers
OBLIQ	Real	None	??	Earth's orbital angle of obliquity (degrees: -90. to +90., typically 22. to 26.). **Setting is Required if IYEAR_AD not set. Not used when running as part of CCSM.**
OMPNEST	Integer	0		Option for nested openmp (used with Finite-Volume dynamical core only).
OZNCYC	Logical	.TRUE.		Flag for yearly cycling of ozone data. If set to .FALSE., a multi-year dataset is assumed, otherwise a single-year dataset is assumed, and ozone will be cycled over the first 12 values in the file.
PERPETUAL_RUN	Logical	false		Set to .true. to specify that the run will use a perpetual calendar. If perpetual_ymd is not set then the perpetual date will be read from the initial file.
PERPETUAL_YMD	Logical	false		Perpetual date specified as (year*1000 + month*100 + day). This date overrides the date from the initial file. If aqua_planet=.true. then perpetual_ymd is ignored and the perpetual date is set to 321.
PERTLIM	Real	0.0		Perturb the initial conditions for temperature randomly by up to the given amount. Only applied for initial simulations.
PHYS_ALLTOALL	Integer	0		Dynamics/physics transpose option.
PHYS_CHNK_PER_THD	Integer	1		Select target number of chunks per thread. Must be positive
PHYS_LOADBALANCE	Integer	0		Select different options for organization of physics chunks. Each uses a different scheme for static load balancing.
PRECC_THRESH	Real	0.1 mm/hr		Precipitation threshhold for use with the PRECCINT and PRECCFRQ fields for history files.
PRECL_THRESH	Real	0.05 mm/hr		Precipitation threshhold for use with the PRECLINT and PRECLFRQ fields for history files.
PRESCRIBED_SULFUR	char*16	direct		Control radiative interaction of prescribed sulfur. May be set to "passive" or "direct". passive = prescribed sulfur has no radiative interaction direct = prescribed sulfur drives radiative interaction

Name	Type	Default	Ex.	Description
PRINT_STEP_COST	Logical	.FALSE.		If true, print CPU timing per model timestep.
PROGNOSTIC_ICESNOW	Logical	.TRUE.		Accumulate snow over sea-ice (with a 0.5m maximum). If .FALSE. then use a climatology for snow depth.
PROGNOSTIC_SULFUR	char*16	off		Control prognostic sulfur calculations. May be set to "off", "passive", or "direct". off = no prognostic sulfur (default) passive = prognostic sulfur computed with no radiative interaction direct = prognostic sulfur computed and drives radiative interaction
RADFORCE	Logical	false		Compute forcing from aerosols
RAMPYEAR_GHG	integer	0		Ramped gases fixed at this year if set to a value greater than zero. Only used if SCENARIO_GHG is set to RAMPED.
RAMPYEAR_PRESCRIBED_SULFUR	integer	none		This variable is reserved for future use. No value is permitted.
RAMPYEAR_PROGNOSTIC_SULFUR	integer	none		Set to YYYY in order to cycle that year of sox emissions.
RAMPYEAR_SCON	integer	0		Ramped scon fixed at this year if set to a value greater than zero. Only used if SCENARIO_SCON is set to RAMPED.
RAMP_CO2_ANNUAL_RATE	real	1.0		percentage amount of co2 ramping per year
RAMP_CO2_CAP	real	See Desc.		co2 cap if rate > 0, floor otherwise; specified as multiple or fraction of inital value; *e.g.* setting to 4.0 will cap at 4x initial co2 setting; default is boundless if rate > 0, zero otherwise
RAMP_CO2_START_YMD	integer	0		date on which ramping of co2 begins; REQUIRED to be set for scenario_ghg='RAMP_CO2_ONLY'
READTRACE	Logical	.TRUE.		If true, initialize data for all consituents found on the initial conditions dataset. Otherwise data will be initialized using internally-specified default values.
REF_TOD	Integer	See Desc.		Reference time-of-day (seconds). Default is to use starting time of day

Name	Type	Default	Ex.	Description
REF_YMD	Integer			Reference date for run as yyyymmdd. Default is to use starting date. The reference date is the reference for the time variable output on the history files.
RESET_CSIM_ICE_PROPS	Logical	.FALSE.		Reset the CSIM ice-model properties. Snow-cover is set to zero, and ice-temperatures are all set to freezing. This is important when starting from an initial dataset that isn't in equilibrium. This is used, for example, when starting from initial conditions that are inter-polated from a different resolution.
REST_PFILE	Char*256	See Desc.		Full pathname of restart pointer file. Default is $HOME/ cam2.CASEID.rpointer
SCENARIO_CARBON_SCALE	char*16	FIXED		Controls carbon scaling. May be set to 'FIXED' or 'RAMPED'. FIXED means use the value specified by CARSCL. RAMPED means use data from file specified by BNDTV-CARBONSCALE.
SCENARIO_GHG	char*16	FIXED		Controls treatment of specified co2,ch4,n2o,cfcf11,cfc12 volume mixing ratios. May be set to 'FIXED' or 'RAMPED' or 'RAMP_CO2_ONLY'. FIXED = volume mixing ratios are fixed and have either default or namelist input values RAMPED = volume mixing ratios are ramped RAMP_CO2_ONLY = only co2 mixing ratios are ramped
SCENARIO_PRESCRIBED_SULFUR	char*16	FIXED		Controls prescribed sulfur amounts. Can only be set to 'FIXED'! FIXED = uses climatology
SCENARIO_PROGNOSTIC_SULFUR	char*16	RAMPED		Sets so2 and so4 surface fluxes. May only be set to 'RAMPED'! RAMPED = uses boundary data set bndtvsox. Only used if PROGNOSTIC_SULFUR is set to PASSIVE or DIRECT.
SCENARIO_SCON	char*16	FIXED		Controls the solar constant. Can be set to 'FIXED' or 'RAMPED'. FIXED = scon is fixed and can have either the default or namelist value. RAMPED = scon is ramped.
SCON	Real	1.367E6	??	Solar constant (W/m^2).
SOIL_EROD	Char*256	See Desc.		Full pathname of time-variant boundary dataset for soil erodibility factors. Default is set in DefaultCAMEXPNamelist.xml.
SOM_CONSCHK_FRQ	Integer	0		This is used only when running with the slab ocean model. If n > 0, SOM global energy checking will be done every n timesteps. If n < 0, SOM global energy checking will be done every n days.

Name	Type	Default	Ex.	Description
SSLTSCL_RF	Real	0		Set sea salt aerosol scaling factor for radiative forcing calculation. This does not affect mmr's used for the climate integration, and is only used when RADFORCE is true.
SSLTSCL	Real	1.0		Set sea salt aerosol scaling factor for radiative transfer calculations. This may be used to scale the specified or prognostic aerosol mass mixing ratios.
SSTCYC	Logical	.TRUE.	??	Flag for yearly cycling of SST data. If set to .FALSE., a multi-year dataset is assumed, otherwise a single-year dataset is assumed, and SSTs will be cycled over the first 12 values in the file. **Not used if running with CCSM.**
START_TOD	Integer	See Desc.		Starting time-of-day (seconds). Default is to use time of day from initial condition dataset. If not input, value is set to NCSEC from initial dataset header.
START_YMD	Integer	See Desc.		Starting date for run as yyyymmdd. Default is to Use date from initial condition dataset. If not input, value is set to NCDATE from initial dataset header.
STOP_TOD	Integer	0		Stopping time of day for run in seconds since 0Z. Not used when running through flux coupler.
STOP_YMD	Integer	None		Stopping date for run encoded in yearmmdd format. Either NELAPSE, NESTEP or STOP_YMD (and possibly STOP_TOD) must be set if not running through flux coupler. (STOP_YMD,STOP_TOD) takes precedence if set. STOP_YMD is not required or used when running as part of CCSM. **Setting is Required if NELAPSE and NESTEP not set.**
STRAT_VOLCANIC	Logical	false		Use stratospheric volcanic aerosols masses and couple with radiative forcing computations.
SULSCL_RF	Real	0		Set sulfur aerosol scaling factor for radiative forcing calculation. This does not affect mmr's used for the climate integration, and is only used when RADFORCE is true.
SULSCL	Real	1.0		Set sulfur aerosol scaling factor for radiative transfer calculations. This may be used to scale the specified or prognostic aerosol mass mixing ratios.
SWAP_COMM_ORDER	Integer	0		Performance tuning option for swap communication. (only used with EUL and SLD dycores).
SWAP_COMM_PROTOCOL	Integer	0		Performance tuning option for swap communication. (only used with EUL and SLD dycores).

Name	Type	Default	Ex.	Description
TAUBACK	Real	0.0		Optical depth of (rh = 0.8, sulfate-like) background aerosol.
TAUVIS	Real	0.14		Visible optical depth.
TRACERTRANS	Integer	0		Number of simultaneously transposed tracers (used with Finite-Volume dynamical core only).
TRACE_GAS	Logical	false		If true, turn on prognostic greenhouse gas calculations for CH4, N2O, CFC11 and CFC12. This requires adding the option "-nadv 7" when running `configuration`.
USE_ETA	Logical	.FALSE.		If TRUE, use the vertical coordinate (eta) values defined in the finite volume dynamical core code rather than the values defined on the input dataset (used with Finite-Volume dynamical core only).
VOLCSCL_RF	Real	0		Set volcanic aerosol scaling factor for radiative forcing calculation. This does not affect mmr's used for the climate integration, and is only used when `RADFORCE` is true.
VOLCSCL	Real	1.0		Set volcanic aerosol scaling factor for radiative transfer calculations. This may be used to scale the specified or prognostic aerosol mass mixing ratios.

Appendix C

CLM namelist variables

This appendix describes a small subset of the namelist variables recognized by CLM 3.0. For more information, please see the CLM User's Guide.

A CAM model run is controlled using the `build-namelist` facility described in Section 2.2. The focus of this appendix is to provide a reference for select CLM variables that may be set through the use of `build-namelist`'s `-namelist` option.

Table C.1: **CLMEXP namelist variables**

Name	Type	Default	Ex.	Description
FINIDAT	Char*256	None		Initial dataset of the land surface model state (initial temperatures, soil water content etcetera). If this dataset is not provided the model will use reasonable values to start from. Note, that some features of the land-surface model take several simulation years to "spin-up" before they reach equilibrium.
FPTFTCON	Char*256	None		Dataset of plant-function types. **Setting is Required**
FSURDAT	Char*256	None		Time invariant surface dataset for this resolution. If this dataset is not provided on an initial run the high resolution datasets MKSRF_FVEGTYP, MKSRF_FSOITEX, MKSRF_FSOICOL, MKSRF_FLANWAT, MKSRF_FURBAN, MKSRF_FGLACIER, and MKSRF_FLAI must be provided, and will be used to create the surface dataset. On a continuation run (branch or restart) this dataset MUST be provided.
MKSRF_FVEGTYP	Char*256	None		High resolution Vegetation Type dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)
MKSRF_FSOITEX	Char*256	None		High resolution Soil Texture dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)

Name	Type	Default	Ex.	Description
MKSRF_FSOICOL	Char*256	None		High resolution Soil Color dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)
MKSRF_FLANWAT	Char*256	None		High resolution Land water (lake and river) dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)
MKSRF_FURBANP	Char*256	None		High resolution Urban Area dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)
MKSRF_FGLACIER	Char*256	None		High resolution Glacier dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)
MKSRF_FLAI	Char*256	None		High resolution Leaf Area Index (LAI) dataset needed to determine the time-invariant surface dataset (FSURDAT). (Required if FSURDAT not defined)

Appendix D

Details of the configuration files

The `configure` utility creates the header files `misc.h`, `params.h`, and `preproc.h` and the directory search path file `Filepath`. These files are written to the directory given by the `-cam_bld` argument of `configure` (the default is the directory from which `configure` is executed). It is not necessary to edit these files to produce supported CAM configurations as that can be easily accomplished by the `configure` utility. However, a user experimenting with non-standard configurations may need to edit these files directly. Thus, the file contents are summarized below.

D.1 `Filepath`

Contains a list of directories used by `gmake` to determine the list of source files to build. All source files listed by the filename expansion expressions `*.F90`, `*.F`, and `*.c` in each of the directories are included in the build. It a given filename occurs in more than one of the directories, it is the version in the first directory of the list that will be built. The list of directories in `Filepath` will vary with target architecture and depend on the specified dynamics and physics packages.

D.2 `misc.h`

Contains a list of resolution-independent `cpp` directives. The following `cpp` tokens must either be defined or undefined (i.e. preceded by `#define` or `#undef`).

D.3 `params.h`

Contains a list of resolution-dependent `cpp` directives for the atmosphere model.

D.4 `preproc.h`

Contains a list of resolution-dependent `cpp` tokens for the land-model code. Currently the resolution of the land-model **must** be the same as that of the atmospheric component.

Table D.1: `misc.h` **pre-processor tokens**

token	Synopsis
COUP_CSM	Define if the flux-coupled ocean configuration will be used. COUP_CSM **is always undefined for stand-alone mode**.
PERGRO	Define if you want to make error growth tests with your model simulation. This token turns off parts of the prognostic cloud water parameterization so that the error growth happens at a reasonable rate. Default setting is with PERGRO undefined.
USEFFTLIB	Define if you want to use a intrinsic FFT package rather than the ECMWF FFT package provided with the code in the "models/atm/cam/src/utils" directory. Default setting is USEFFTLIB undefined.
SPMD	Enables the distributed memory (SPMD) implementation.
STAGGERED	Enables staggered grid processing. Required for use with the Lin-Rood dynamical core, otherwise undefined.

D.5 config_cache.xml

Not yet described.

D.6 Makefile

Not yet described.

Table D.2: `params.h` **pre-processor tokens**

token	**Synopsis**
`PCNST`	Number of advected constituents including water vapor and cloud-water. Default setting is `1` (for water vapor).
`PNATS`	Number of non-advected constituents. Default setting is `1` (for cloud water).
`PLEV`	Number of vertical levels. Default setting is 26.
`PLEVR`	Number of vertical levels over which radiation calculations are performed. Use of separate vertical coordinate for the radiation calculation has not been tested. Therefore this directive must currently be set to the same value as `PLEV`.
`PLON`	Number of longitudes on the transform grid. Default setting is 128.
`PLAT`	Number of Gaussian latitudes on the CAM2.0 transform grid. Default setting is 64.
`PCOLS`	Maximum number of columns to use for the physics. This value can be tuned for different computer architectures in order to increase performance. Default setting is 16.
`PTRM`	Spectral truncation of the zonal wavenumber `m`. Default setting is 42.
`PTRN`	Spectral truncation of the total wavenumber `n` for zonal wavenumber 0. Default setting is 42.
`PTRK`	Maximum total wavenumber `k`, for any zonal wavenumber `m`. Default setting is 42.

Table D.3: `preproc.h` **pre-processor tokens**

token	**Synopsis**
`COUP_CAM`	Tell the CLM that it is being run as a subroutine beneath the atmospheric model.
`LSMLON`	Number of longitudes for the landmodel grid. Currently this directive must have the same value as `PLON` in `params.h`.
`LSMLAT`	Number of latitudes for the land-model grid. Currently this directive must have the same value as `PLAT` in `params.h`.

Bibliography

Collins, W. D., P. J. Rasch, and Others, Description of the NCAR Community Atmosphere Model (CAM 3.0), Technical Report NCAR/TN-464+STR, National Center for Atmospheric Research, Boulder, Colorado, 210 pp., 2004.