Community Climate System Model
National Center for Atmospheric Research, Boulder, CO

CLM2.0 User's Guide

Mariana Vertenstein, Keith Oleson and Sam Levis

May 2002

Contents

1 Obtaining the Source Code and Datasets

The source code and datasets required to run the Community Land Model version 2.0 (CLM2.0) in offline mode (uncoupled from other components of the Community Climate System Model version 2 (CCSM2.0)) can be obtained via the web from:

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

The user should refer to the CAM2.0 User's Guide or the CCSM2.0 User's Guide for instructions on obtaining code and datasets to run CLM2.0 coupled to other CCSM2.0 components.

It is assumed that the user has access to the utilities **tar**, Free Software Foundation **gunzip** and **gmake (GNU gmake)**.

The CLM2.0 distribution consists of two tar files:

CLM2.0_code.tar.gz

and

CLM2.0_inputdata.tar.gz.

The file CLM2.0_code.tar.gz contains code, documentation, and scripts. This file must first be uncompressed with the **gunzip** utility and then "untarred" as follows:

gunzip -c CLM2.0_code.tar.gz | tar xvf -

The above command both uncompresses and "untars" the code into a **clm2**/ subdirectory. The directory hierarchy for "**clm2**/" is as follows:

Directory Name	Synopsis
src/	Directory of FORTRAN and "C" source code
src/biogeophys/	Biogeophysics routines (e.g., surface fluxes)
src/camclm_share/	Code shared between the CLM2 and CAM (e.g., calendar information)
src/csm_share/	Code shared by all the geophysical model components of the Community Climate System Model (CCSM). Currently contains code for CCSM message passing orbital calculations and system utilities
src/ecosysdyn/	Ecosystem dynamics routines (e.g., leaf and stem area index)
src/main/	Control (driver) routines
src/mksrfdata/	Routines for generating surface datasets
src/riverroute/	River routing (RTM) routines
src/utils/	Independent utility routines
src/utils/esmf/	Earth System Modeling Framework utilities
src/utils/timing/	General purpose timing library
bld/	Directory of build, test and run scripts
bld/offline/	Script to build and execute the model on various platforms
bld/offline/tests	Perl scripts for model development testing
doc/	CLM2.0 documentation
tools/	Directory of tools for input dataset manipulation
tools/convert-ascii/	Routines for converting user-generated ascii surface dataset files to netCDF format suitable for use by the model (this tool is used independent of running the model)

tools/cprlndnc/	Produces executable that compares CLM2.0 netCDF history files (this tool is used independent of running the model)
tools/makdep/	Produces executable needed by Makefile to generate dependencies when the model is built.

The file CLM2.0_inputdata.tar.gz contains surface and offline atmospheric forcing datasets. This file must first be uncompressed with the **gunzip** utility and then "untarred" as follows:

gunzip -c CLM2.0_inputdata.tar.gz | tar xvf -

The above command both uncompresses and "untars" the file into a **inputdata/lnd/clm2/** subdirectory. The directory hierarchy for "**inputdata/lnd/clm2/**" is as follows:

Directory Name	**Synopsis**
NCEPDATA/	One year's worth of atmospheric forcing variables in monthly netCDF format suitable for running the model in offline mode (uncoupled from the atmospheric model)
inidata/	Directory heirarchy containing netCDF CLM2.0 initial datasets
inidata/cam	Initial datasets for initializing CLM2.0 from a spun-up state when running in cam mode (can also be used when running in offline mode)
inidata/csm	Initial datasets for initializing CLM2.0 from a spun-up state when running in csm mode (can also be used when running in offline mode)
inidata/offline	initial datasets for initializing CLM2.0 from a spun-up state when running in offline mode
pftdata/	Plant functional type (PFT) physiological constants dataset (ascii format)
rawdata/	"Raw" (highest provided resolution) datasets (netCDF format) (used by CLM2.0 to generate surface datasets at model resolution)
rtmdata/	River direction map for RTM in ascii format
srfdata/	Directory heirarchy containing netCDF CLM2.0 surface datasets
srfdata/cam	Surface datasets for running CLM2.0 in cam mode (can also be used when running in offline mode)
srfdata/csm	Surface datasets for running CLM2.0 in csm mode (can also be used when running in offline mode)
srfdata/offline	Surface datasets for running CLM2.0 in offline mode

2 Creating and Running the Executable

The CLM2.0 model can be built to run in one of three modes. It can run as a stand alone executable where atmospheric forcing data is periodically read in (e.g., using the data in **NCEPDATA**). This will be referred to as offline mode. It can also be run as part of the Community Atmosphere Model (CAM) where communication between the atmospheric and land models occurs via subroutine calls. This will be referred to as cam mode. Finally, it can be run as a component in a system of geophysical models (CCSM). In this mode, the atmosphere, land, ocean and sea-ice models are run as separate executables that communicate with each other via the CCSM flux coupler. This will be referred to as csm mode.

The following table lists the supported target architectures for the different modes:

Mode	SUPPORTED Platforms
offline	IBM(SP) (AIX), SGI (IRIX64)
cam	IBM(SP) (AIX), SGI (IRIX64), Linux, Compaq (OSF1), Sun (SunOS)
csm	IBM(SP) (AIX), SGI (IRIX64)

The IBM(SP) is a distributed memory machine consisting of multiple compute nodes. Each node in turn contains multiple shared memory processors (currently four) and network connections which attach it to the other nodes. When running on the IBM(SP), CLM2.0 uses Open Multi Processing (OpenMP) directives within a single (shared memory) node and Message Passing Interface (MPI) directives across nodes (distributed memory) to take full advantage of parallelism within and across several nodes.

The SGI is a shared memory RISC architecture machine. The optimal method of running CLM2 on the SGI is through use of only OpenMP directives to take advantage of shared memory parallelism. If OpenMP directives are used (default), MPI should not be invoked (the cpp directive SPMD should not be defined).

The method of building and running CLM2.0 depends on the selected mode as well as the target architecture. A general discussion of the various aspects of building and running CLM2.0 follows.

2.1 Offline mode: Using jobscript.csh

In order to build and run CLM2.0 in offline mode, a sample script, jobscript.csh, and a corresponding Makefile are provided in the **bld/offline** directory. The script creates a model executable at 3x3 resolution, determines the necessary input datasets, constructs the input model namelist and runs the model for one day.

The user must edit this script appropriately in order to build and run the executable for their particular requirements. This script is provided only as an example to help the novice user get CLM2.0 up and running as quickly as possible.

Jobscript.csh can be run with minimal user modification, assuming the user resets several environment variables at the top of the script. In particular, the user must set **MODEL_DATADIR** to point to the full disk pathname of the directory containing the untarred input data subdirectories **NCEPDATA**, **inidata**, **pftdata**, **rawdata**, **rtmdata** and **srfdata**. The user must also set **MODEL_SRCDIR** to point to the full disk pathname of the directory containing the untarred source code subdirectories : **biogeophys**, **camclm_share**, **csm_share**, **ecosysdyn**, **ecosysdyndgvm**, **main**, **mksrfdata**, **riverroute** and **utils**. Finally, the user must set **MODEL_EXEDIR** to point to the directory where the user wants the executable to be build and run.

The script can be divided into five functional sections: 1) specification of script environment variables; 2) creation of two header files (misc.h and preproc.h) and a directory search path file (Filepath) needed to build the model executable; 3) creation of the model input namelist; 4) creation of the model executable; and 5) execution of the model. Each of these functional sections is discussed in what follows.

2.1.1 Specification of environment variables

The following environment variables are set from within the script. The script provides tentative settings for all variables, however, these values MUST BE edited by the user.

Environment Variable	Synopsis
MODEL_SRCDIR	Full pathname for the source code directory hierarchy
MODEL_EXEDIR	Full pathname for the directory where model executable will reside Object files will be built in the directory $MODEL_EXEDIR/obj

MODEL_DATADIR	Full pathname for the directory where the input datasets reside
DEBUG	Turns debugging flags on in Makefile if set
NTHRDS	Number of OpenMP multitasking threads Should not exceed the number of physical CPUs (ie, processors) on a shared memory machine Should not exceed the number of CPUs in a node on a distributed memory machine
NTASKS	Number of MPI tasks to use for distributed memory implementation If NTASKS = 1, distributed memory is disabled If NTASKS > 1, distributed memory is enabled on NTASKS MPI processes
LIB_NETCDF	Full pathname for directory containing the netCDF library Setting depends on user's target architecture
INC_NETCDF	Full pathname for directory containing netCDF include files Setting depends on user's target architecture
LIB_MPI	Full pathname for directory containing the MPI library Setting depends on user's target architecture Only needed if NTASKS > 1
INC_MPI	Full pathname for directory containing the MPI include files Setting depends on user's target architecture Only needed if NTASKS > 1

2.1.2 Creation of header files and directory search path

The script creates the header files misc.h and preproc.h and the directory search path file Filepath. These files are placed in the directory $MODEL_EXEDIR/obj. To modify these files the user should edit their contents from within the script rather than attempt to edit the files directly since the script will overwrite the files upon its execution. The use of these files by **gnumake** is discussed in section 2.1.4. The contents of each of these files are summarized below.

The file misc.h contains a list of resolution- and model-independent C-language pre-processor (cpp) tokens

misc.h cpp token	**Synopsis**
SPMD	If defined, enables distributed memory (single program multiple data (SPMD)) implementation (Automatically defined if environment variable NTASKS > 1)
PERGRO	If defined, enables modification that tests reasonable perturbation error growth Only applicable in cam mode

The file, preproc.h, contains a list of resolution- and model-dependent cpp tokens.

preproc.h cpp token	**Synopsis**
OFFLINE	If defined, offline mode is invoked
COUP_CSM	If defined, csm mode is invoked
COUP_CAM	If defined, cam mode is invoked
LSMLON	Number of model longitudes
LSMLAT	Number of model latitudes
RTM	If defined, RTM river routing is invoked

The file Filepath contains a list of directories used by **gnumake** to resolve the location of source files and to determine dependencies. Users can add new search directories by editing jobscript.csh under "build Filepath". The default Filepath directory hierarchy for CLM2 is as follows:

Source Directories	**Functionality**
$MODEL_SRCDIR/main	control routines (history, restart, etc)
$MODEL_SRCDIR/biogeophys	biogeophysics routines
$MODEL_SRCDIR/ecosysdyn	ecosystem dynamics routines
$MODEL_SRCDIR/riverroute	river routing routines
$MODEL_SRCDIR/camclm_share	code shared between CAM and CLM2
$MODEL_SRCDIR/csm_share	code shared by all CCSM geophysical model components
$MODEL_SRCDIR/utils/timing	timing routines
$MODEL_SRCDIR/mksrfdata	generation of surface dataset routines

2.1.3 Setting the Namelist

Before building and running the model, the user must specifiy CLM2 namelist variables appearing in the CLM2 namelist, **clmexp**. A default namelist is generated by jobscript.csh. This namelist will result in a one day model run using the provided datasets. Namelist input is written to the file lnd.stdin and can be divided into several main categories: run definitions, datasets, history and restart file settings, and land model physics settings. A full discussion of possible namelist settings is given in Section 3.

2.1.4 Building the model

The script, jobscript.csh, invokes **gnumake** to build (compile) the model. The file, Makefile, located in the **bld/offline** directory, contains the commands used by **gnumake** that are required for each of the supported target architectures. The executable name resulting from the build procedure is "clm". The result of the build procedure will be documented in the log file, compile_log.clm. Any problems encountered during the build procedure will be documented here.

Gnumake generates a list of source and object files using each directory listed in Filepath. For each source file, **gnumake** invokes cpp to create a dependency file in the directory $MODEL_EXEDIR/obj. For example, routine.F90 will have a dependency file, routine.d. If a file which is listed as a target of a dependency does not exist in $MODEL_EXEDIR/obj, **gnumake** 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, the directory containing the user code.

A parallel **gnumake** is achieved in the script by invoking **gnumake** with the -j option, which specifies the number of job commands to run in parallel.

To obtain a model executable, the environment variables LIB_NETCDF and INC_NETCDF must be specified. These provide pathnames to netCDF library and include files. Furthermore, if CLM2.0 is run under MPI (the environment variable NTASKS is greater than 1 in the script and the C-preprocessor directive SPMD is defined), then the directories containing the MPI library and MPI include files must also be specified as environment variables in the script. (This is not the case only for the IBM-SP, where the MPI library and include files are obtained directly from choice of compiler command).

C-preprocessor directives of the form #include, #if defined, etc., are used to enhance code portability and allow for the implementation of distinct blocks of functionality (such as incorporation of different modes) within a single file. Header files, such as misc.h, are included with #include statements within the source code. When **gnumake** is invoked, the C preprocessor includes or excludes blocks of code depending on which cpp tokens have been defined. C-preprocessor directives are also used to perform textual substitution for resolution-specific parameters in the code. The format of these 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(lsmlon = LSMLON); parameter(lsmlat = LSMLAT)

will result in the following processed line (for 3x3 model resolution):

parameter(lsmlon = 120) ; parameter(lsmlat = 60)

where LSMLON and LSMLAT are set in preproc.h via the jobscript.

2.1.5 Running the executable

Jobscript.csh will execute the commands required to run the model under the supported target architectures. The settings of the environment variables NTASKS and NTHRDS in the script determine the CLM2.0 runtime environment. If NTHRDS is set to greater than 1, OpenMp multitasking will be used for the number of threads specified. If NTASKS is greater than 1, CLM2.0 will be run under MPI for the number of tasks specified. If both are greater than 1 (this should only be used for the IBM(SP)), then hybrid mode OpenMP/MPI will be enabled.

If MPI is used for model execution, most MPI implementations provide a startup script which accepts the MPI executable as a command line argument. Additional command line arguments allow the user to specify details such as the various machine architectures or number of processes to use for the run. Once MPI has created the specified number of processes, model execution will begin. The collection of active tasks will then compute locally and exchange messages with each other to integrate the model.

Upon successful completion of the model run, several files will be generated **in the executable directory**. These include history, restart, and initialization files (see section 3.3 for more details), as well as log files that document the execution of the model. The log files will be located in the directory corresponding to the script environment variable, $MODEL_EXEDIR, and will have the form, clm.log.YYMMDD-HHMMSS, where YY is the last two digits of the current year, MM is the month, DD is the day of the month, HH is the hour, MM is the minutes, and SS is the seconds of the start of the model run. These logs file may be referred to as "standard out." A timing file, timing.0, containing model performance statistics is also generated in the executable directory.

2.2 Cam mode

When running CLM2.0 as part of the CAM executable, CAM build and run scripts must be utilized and the user should refer to the CAM User's Guide for specific details on building and running the CAM executable. We will only discuss some essential points of the CAM build and run scripts.

The header files, preproc.h and misc.h, as well as the directory search path file, Filepath, are needed for the CAM build procedure in an analogous manner to the CLM2.0 build procedure. The user should keep in mind that the CLM2.0 directory hierarchy **MUST appear after** the CAM directory hierarchy in Filepath. CLM2.0 contains several files that have the same name as the corresponding CAM files (e.g. time_manager.F90). When running in CAM mode, the corresponding CAM file must be used. The CAM build and run scripts ensure this.

The CLM2.0 namelist, **clmexp**, must also be specified. By default, RTM river routing is not enabled in cam mode (i.e. the cpp variable, RTM, is not defined). Furthermore, CLM2.0 does not permit the user to independently set several namelist variables (in particular, those dealing with history file logic and run control logic) when running in cam mode. CLM2.0 will override any user input for these variables with the corresponding values used by the CAM model. This is discussed in more detail in section 3.6.

2.3 Csm mode

When running CLM2.0 as the land component of CCSM, CCSM build and run scripts must be utilized. The user should refer to the CCSM2.0 Quick Start Guide and associated documentation for a complete description. We will only briefly outline some of the key points associated with executing CLM2.0 as part of CCSM2.0.

The master CCSM script, test.a1.run, coordinates the building/running of the complete system of CCSM executables. The land component setup script, lnd.setup.csh (equivalent to jobscript.csh in the offline case), builds the CLM2.0 executable and creates the input CLM2.0 namelist, lnd.stdin. The ending time step need not be specified in the CLM2.0 namelist since the model responds to flags set in the flux coupler script, cpl.setup.csh, that determine when it is time to terminate the run. For a full discussion of CLM2.0 namelist varibles, csm mode included, refer to section 3.

3 Namelist or Resource File Parameters

CLM2.0 namelist inputs are presented in several tables below. In these tables, "mode" has values of "offline", "csm", "cam" or "all", corresponding to offline mode, csm mode, cam mode, or all the modes. If a namelist variable setting is listed as **REQUIRED**, then the value must be set in the namelist in order for CLM2.0 to execute successfully. If a setting is specified as **REQUIRED** and the mode is only given as offline, then that variable must only be specified when running in offline mode. For namelist variable settings not listed as **REQUIRED**, the code will provide default settings at initialization. In the following variable descriptions, we refer to examples presented in Section 5.

3.1 Specification of run length, run type and initial run date

The following table specifies namelist variables associated with the definition of run case names, types (restart, initial or branch), model time step, and initial run date. An initial run starts the model from either arbitrary initial conditions that are set internally in the code or using an initial conditions dataset (see namelist variable **FINIDAT**) that enables the model to start from a spun-up state. A restart run is an exact continuation of a previous simulation from its point of termination. Output from a restart run should be bit-for-bit the same as if the previous simulation had not stopped. Run control variables set in the namelist must be the same as in the run that is being restarted. A branch run is a new case that uses restart data from a previous simulation to begin the integration. For a branch run, 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 original run.

Namelist: Define the type and length of run	
CASEID	
description	case name (short identifier for run) (see ex. 3)
type	char*32
mode	offline, csm (obtained from atm in cam mode)
default	**REQUIRED** (must be changed for branch run)
CTITLE	
description	case title for use within history files (long identifier)
type	char*80
mode	offline, csm (obtained from atm in cam mode)
default	blank
NSREST	
description	run type (0 for initial run, 1 for restart, 3 for branch) (see ex. 1)
type	integer
mode	offline, csm (obtained from atm in cam mode)
default	**REQUIRED**

DTIME	
description	model time step (seconds) (see ex. 1)
type	real
mode	offline, must agree with CAM in csm mode, obtained from atm in cam mode
default	**REQUIRED** (suggested range: 1200-3600 s)
NELAPSE	
description	elapsed run time in model time steps (positive) or days (negative) (see ex. 2)
type	integer
mode	offline (obtained from atm/coupler in cam/csm mode, respectively)
default	**REQUIRED** (if NESTEP not set)
NESTEP	
description	ending run time in model time steps (positive) or days (negative) (see ex.ample 1)
type	integer
mode	offline (obtained from atm/coupler in cam/csm mode, respectively)
default	**REQUIRED** (if NELAPSE not set)
START_YMD	
description	start date of run (yyyymmdd format) (see ex. 1)
type	integer
mode	offline, csm (obtained from atm in cam mode)
default	**REQUIRED**
START_TOD	
description	start time of day of run (seconds) (see ex. 1)
type	integer
mode	offline, csm (obtained from atm in cam mode)
default	0

3.2 Specification of model input datasets

The following table specifies namelist variables associated with model input datasets.

Namelist: Input datasets	
FINIDAT	
description	full pathname of initial conditions dataset (see ex. 4)
type	char*256
mode	all
default	blank
notes	datasets provided are in $MODEL_DATADIR/inidata
FSURDAT	
description	full pathname of surface dataset (see ex. 1)
type	char*256
mode	all
default	blank
notes	raw datasets to generate surface dataset provided are in $MODEL_DATADIR/rawdata surface datasets provided with the distribution are in $MODEL_DATADIR/srfdata
FPFTCON	
description	full pathname of plant functional type (PFT) physiological constants dataset (see ex. 1)
type	char*256
mode	all

default	**REQUIRED**
notes	dataset provided is $MODEL_DATADIR/pftdata/pft-physiology
FRIVINP_RTM	
description	full pathname of RTM input dataset (see ex. 4)
type	char*256
mode	offline, csm
default	**REQUIRED** if cpp token **RTM** is defined in preproc.h
notes	dataset provided is $MODEL_DATADIR/rtmdata/rdirc.05
NREVSN	
description	full pathname of restart file name (only for branch runs) (see ex. 3)
type	char*256
mode	all
default	**REQUIRED** (only if branch run, NSREST=3)
MKSRF_FVEGTYP	
description	full pathname of raw vegetation type dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_pft.nc
MKSRF_FSOITEX	
description	full pathname of raw soil texture dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_soitex.10level.nc
MKSRF_FSOICOL	
description	full pathname of raw soil color dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	$MODEL_DATADIR/rawdata/mksrf_soicol_clm2.nc
MKSRF_FLANWAT	
description	full pathname of raw inland water dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_lanwat.nc
MKSRF_FURBAN	
description	full pathname of urban dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_urban.nc
MKSRF_FGLACIER	
description	full pathname of glacier dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_glacier.nc
MKSRF_FLAI	

description	full pathname of leaf and stem area index, canopy top and bottom height dataset (see ex. 5)
type	char*256
mode	all
default	**REQUIRED** (if FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_lai.nc
MKSRF_OFFLINE_FNAVYORO	
description	20 min navy orography dataset used to generate land mask (see ex. 5)
type	char*256
mode	offline
default	**REQUIRED** (if MKSRF_OFFLINE_FGRID not set and FSURDAT is blank)
notes	dataset provided is $MODEL_DATADIR/rawdata/mksrf_navyoro_20min.nc
MKSRF_OFFLINE_FGRID	
description	dataset specifying land grid and mask at desired resolution
type	char*256
mode	offline
default	blank, **REQUIRED** (if MKSRF_OFFLINE_FNAVYORO not set and fsurdat is blank)
MKSRF_OFFLINE_EDGEN	
description	northern edge of land grid (degrees north) (see ex. 5)
type	real
mode	offline
default	90.
MKSRF_OFFLINE_EDGEE	
description	eastern edge of land grid (degrees east) (see ex. 5)
type	real
mode	offline
default	180.
MKSRF_OFFLINE_EDGES	
description	southern edge of land grid (degrees north) (see ex. 5)
type	real
mode	offline
default	-90.
MKSRF_OFFLINE_EDGEW	
description	western edge of grid land (degrees east) (see ex. 5)
type	real
mode	offline
default	-180.
OFFLINE_ATMDIR	
description	directory containing atmospheric forcing datasets (see ex. 1)
type	char*256
mode	offline
default	**REQUIRED**
notes	data provided is in directory $MODEL_DATADIR/NCEPDATA

The file, **FSURDAT**, specifies a surface dataset containing time-invariant land properties such as plant functional types and soil textures and time-variant properties such as leaf area index. Several surface datasets are provided with the distribution:

- $MODEL_DATADIR/srfdata/offine: offline mode surface datasets

- $MODEL_DATADIR/srfdata/cam: cam mode surface datasets
- $MODEL_DATADIR/srfdata/csm: csm mode surface datasets

If the value of **FSURDAT** is set to the empty string, a new surface dataset will be generated at run time for the specified model resolution and land/ocean mask. The creation of the new surface dataset requires that the full pathname of the following raw datasets be specified:

- **MKSRF_FVEGTYP**
- **MKSRF_FSOITEX**
- **MKSRF_FSOICOL**
- **MKSRF_FLANWAT**
- **MKSRF_FURBAN**
- **MKSRF_FGLACIER**
- **MKSRF_FLAI**

These raw datasets are only used when a surface dataset is created at run time. The required raw datasets are provided with the distribution and are contained in the directory $MODEL_DATADIR/rawdata.

In addition to raw datasets, a land/ocean mask is required in order for a new surface dataset to be created. If CLM2.0 is run in csm or cam mode, this mask is obtained from either the CCSM flux coupler or from the CAM atmosphere model at run time. In offline mode, however, the land/ocean mask can either be calculated from a high resolution orography dataset by setting the namelist variable **MKSRF_OFFLINE_FNAVYORO** or can be read in from a dataset via the setting of the namelist variable **MKSRF_OFFLINE_FGRID**.

Subroutines involved in creating a surface data set from the raw data reside in the directory **$MODEL_SRCDIR/mksrfdata**. In most cases the creation of a surface dataset involves a straightforward interpolation from the raw data resolution to the desired model resolution. For soil texture, however, averaging would create new soil types. Consequently, the model determines the dominant soil texture profile per gridcell from the raw resolution to the desired resolution. Once the surface dataset is created, the CLM2.0 code reads it back in so that the same results will be obtained regardless of whether a run starts from an existing surface dataset or creates one at the same resolution at start up.

The file, **FINIDAT**, contains values for the time-dependent variables needed to initialize CLM2.0 from a spun-up state. If **FINIDAT** is set to the empty string, these variables are initialized to non spun-up values in CLM2.0. **HIST_CRTINIC** described in the next section can be used to create these files during a model run. CLM2.0 initial datasets provided with the distribution can be found in the following directories:

- $MODEL_DATADIR/inidata/offine: offline mode initial datasets
- $MODEL_DATADIR/inidata/cam: cam mode initial datasets
- $MODEL_DATADIR/inidata/csm: csm mode initial datasets

FPFTCON specifies the data file containing plant functional type physiological constants. The dataset provided with the distribution is **$MODEL_DATADIR/pftdata/pft-physiology**.

If the cpp token **RTM** is defined, then the RTM river routing scheme will be invoked in running CLM2.0. In this case, **FRIVINP_RTM** must be set to a river routing dataset. The dataset provided with the distribution is **$MODEL_DATADIR/rtmdata/rdirc.05**

NREVSN is ignored unless a branch run is specified (i.e., **NSREST** is set to 3).

In offline mode, time dependent atmospheric forcing data must be read in. The directory containing these files is given by **OFFLINE_ATMDIR**. The forcing data provided with the distribution can be found in **$MODEL_DATADIR/NCEPDATA**. This variable is ignored in cam or csm mode.

3.3 Specification of history and restart files

The following table specifies namelist variables associated with history, restart, and initialization files.

Namelist: History and restart files	
HIST_CRTINIC	
description	frequency to generate initial dataset that can be used for future initial runs [set to 'MONTHLY','YEARLY','NONE']
type	char*8
mode	offline, csm (obtained from atm in cam mode)
default	'YEARLY'
HIST_NHTFRQ(1)	
description	primary history file interval (+ for model time steps, - for hours, 0 for monthly ave) (see ex. 4)
type	integer
mode	offline, csm (obtained from atm in cam mode)
default	-24
HIST_MFILT(1)	
description	number of time samples per primary history file (see ex. 4)
type	integer
mode	offline, csm (obtained from atm in cam mode)
default	1
HIST_NHTFRQ(2:maxhist)	
description	auxillary history file interval (see ex. 4)
type	integer
mode	all
default	**required** (if auxillary files are requested)
HIST_MFILT(2:maxhist)	
description	number of time samples per auxiliary history file (see ex. 4)
type	integer
mode	all
default	**required** (if auxillary files are requested)
HIST_NDENS	
description	output file precision, 1 (double precision) or 2 (single precision)
type	integer
mode	all
default	1
HIST_DOV2XY(1:maxhist)	
description	true if want grid-average history field as opposed to vector history field (see ex. 4)
type	logical
mode	all
default	true
HIST_CHNTYP(2,maxalflds)	
description	paired field name and field type (i.e., instant, maximum, minimum, average, constant) to override default settings (see ex. 4)
type	char*8
mode	all
default	blank
HIST_FLDAUX1(maxalflds)	
description	names of fields for first auxillary history file (see ex. 4)
type	char*8

mode	all
default	blank
HIST_FLDAUX2(maxalflds)	
description	names of fields for second auxillary history file
type	char*8
mode	all
default	blank
HIST_FLDADD(maxalflds)	
description	names of fields to change to active in primary history file (see ex. 4)
type	char*8
mode	all
default	blank
MSS_IRT	
description	mass store retention period (days) (see ex. 4)
type	integer
mode	offline, csm (obtained from atm in cam mode)
default	0 (i.e., history files will be written to local disk, not the mass store)
MSS_WPASS	
description	mass store write password for output data sets
type	char*8
mode	all
default	blank
RPNTPATH	
description	full unix pathname of the local restart pointer file
type	char*256
mode	all
default	lnd.CASEID.rpointer in home directory

CLM2.0 writes its own history, restart and initial files. History files are in netCDF file format and contain model data values written at specified times during a run. Generally these values are time-averaged. CLM2.0 produces a primary history file and allows the user to define up to two auxiliary history files. Primary history files contain the string 'h0', whereas auxiliary history files contain the string 'h1' or 'h2'.

Initial dataset files are instantaneous netCDF files containing only initial data fields. CLM2.0 produces initial datasets either yearly, monthly, or not at all depending on the setting of the namelist variable **HIST_CRTINIC** (the default setting is "YEARLY"). These datasets can be utilized as "spun-up" initial conditions.

Restart files are in binary format and can be used only to restart or branch runs from previous model simulations. Whenever a restart file is written, a corresponding restart pointer file on local disk (lnd.CASEID.rpointer) is overwritten.

The following table specifies the naming convention used for output files. In this table the string yyyy refers to the model year, mm refers to the model month, dd is the model day and sssss corresponds to seconds into the model day. Note that for non-monthly history files, yyyy-mm-dd-sssss corresponds to the first timestamp of data on the file. **CASEID** is the case identifier set via the namelist input.

File Name	**Synopsis**
CASEID.clm2.r.yyyy-mm-dd-sssss	restart file
CASEID.clm2.i.yyyy-mm-dd-sssss	initial file

CASEID.clm2.h[012].yyyy-mm	monthly average history file (primary or aux)
CASEID.clm2.h[012].yyyy-mm-dd-sssss	non-monthly history file (primary or aux)

Model parameters relating to history files appear in routine **src/main/clm_varpar.F90**. The maximum number of history files is set by the parameter **maxhist**. Currently, there can be at most two auxiliary history files (the dimension **maxhist** is set to 3). The maximum number of fields is set by the parameter, **maxalflds**, whose dimension is defined as the sum of the maximum allowable single-level fields (75) and multi-level fields (10).

History fields are either defined as 'active' or 'inactive'. These default settings are set in routine **src/main/histFileMod.F90**. Default inactive fields can be made active by setting the **HIST_FLDADD** variable to the appropriate field name via the namelist input. All active fields will appear on the primary history file. A field set to inactive cannot appear on auxillary files.

The namelist variables **HIST_FLDAUX1** and **HIST_FLDAUX2** are used to specify desired output fields on auxiliary history files. Time averaging aspects of auxiliary files are controlled via settings of the namelist variables **HIST_NHTFRQ(2:maxhist)** and **HIST_MFILT(2:maxhist)**.

Field time averaging types can be overridden by setting the **HIST_CHNTYP** variable to the appropriate field name. Valid field types are "average" (average field over history interval), "maximum" (maximum field value), "minimum" (minimum field value), "instant" (instantaneous field value) and "constant" (time invariant field value). An example of this specification would be, **HIST_CHNTYP** = 'TSA','maximum'. This specifies that the 2-m air temperature is to be recorded on the primary history file as a maximum value over the specified history interval (e.g., if the history interval is monthly, the recorded value would be the maximum value of the 2-m air temperature occurring within that month).

The default averaging for each field and a list of active fields can be found in the routine **src/main/histFileMod.F90**. The table in section 4 contains a list of all fields currently in **histFileMod.F90**.

History, restart and initialization files can be archived on the NCAR mass storage system (MSS) if the namelist variable, **MSS_IRT**, is set to a value greater than zero. History, restart and initial files are archived as follows (where **USERNAME** is the upper-case equivalent of the user's login name, i.e., the user's root directory on the MSS):

- history files: /USERNAME/csm/CASEID/lnd/hist
- restart files: /USERNAME/csm/CASEID/lnd/rest
- initial files: /USERNAME/csm/CASEID/lnd/init

3.4 Specification of input physics variables

Namelist: Input physics	
CONCHK	
description	does error checks on energy and water conservation (see example 4)
type	logical
mode	all
default	true
IRAD	
description	frequency of solar radiation calculations (+ for model time steps, - for hours) (see ex. 4)
type	integer
mode	offline, must be consistent with CAM in csm mode obtained from atm in cam mode
default	-1
CSM_DOFLXAVE	
description	if true, flux averaging is performed over the duration set in **IRAD**
type	logical
mode	csm (must agree with CAM setting)
default	true
WRTDIA	
description	if true, global average 2-m temperature written to standard out (ascii log file of the run) (see ex. 4)
type	logical
mode	all
default	false

3.5 Specification of RTM River routing

Namelist: RTM River Routing	
RTM_NSTEPS	
description	average rtm output over rtm_nsteps time steps
type	integer
mode	all
default	number of timesteps in 3 hours

3.6 Cam mode namelist specification

When running CLM2.0 in cam mode, certain CLM2.0 namelist variables cannot be set independently. In particular, any user specification for the CLM2.0 namelist variables, **CASEID**, **CTITLE**, **IRAD**, **NSREST**, **HIST_CRTINIC**, **HIST_NHTFRQ(1)**, **HIST_MFILT(1)**, **MSS_IRT** will be overwritten by values obtained from the CAM model. All other namelist settings that are active in cam mode may be set independently by the user.

The following table specifies the CLM namelist variables that are overwritten with CAM values, and lists the associated CAM namelist variable and its default value.

CASEID	
CAM namelist	CASEID
CAM default	REQUIRED
CTITLE	
CAM namelist	CTITLE
CAM default	blank

NSREST	
CAM namelist	NSREST
CAM default	0
IRAD	
CAM namelist	IRADSW
CAM default	-1
HIST_CRTINIC	
CAM namelist	INITHIST
CAM default	'MONTHLY'
HIST_NHTFRQ(1)	
CAM namelist	NHTFRQ(1)
CAM default	0
HIST_MFILT(1)	
CAM namelist	MFILT(1)
CAM default	1
MSS_IRT	
CAM namelist	MSS_IRT
CAM default	365

The minimum set of CLM2.0 namelist variables that must be given values in the input namelist depends on if a pre-existing surface dataset and/or initial dataset will be used and if RTM is enabled. At a minimum, only the namelist variables **FPFTCON** must be given a value if no surface dataset exists and if the cpp variable RTM is not defined. If RTM is defined, then **FRIVINP_RTM** must also be specified. Finally, the namelist variables **FSURDAT** and/or **FINIDAT** must be set appropriately if a pre-existing surface dataset and/or initial datasets are to be utilized.

3.7 Csm mode namelist specifications

When running in csm mode, the user must make sure that the settings of the following namelist variables:

- **IRAD**
- **DTIME**
- **CSM_DOFLXAVE**

have identical values to the corresponding CAM2.0 namelist variables (see CAM2.0 user's guide).

Two input datsets are required in csm mode: an RTM input dataset, rdirc.05, associated with the namelist variable **FRIVINP_RTM** and a dataset containing ecophysiological constants, pft-physiology, associated with the namelist variable **FPFTCON**.

Two other input datasets may also be provided, but are not required. A surface dataset will be generated at run time if the namelist variable **FSURDAT** is not specified. In csm mode, the CAM and CLM2.0 grids must be identical. However, the land mask for this grid is provided by the flux coupler at run time. This land mask is dependent on the specific ocean domain being used. Consequently, a different surface dataset is required for each atm/ocn grid combination. Surface dataset names are automatically generated by lnd.setup.csh given the corresponding atm/ocn resolutions. Currently, two standard combinations are supported: [T42 atm] with [gx1v3 ocn] and [T31 atm] with [gx3 ocn].

Additionally, a spun-up initial dataset may be provided containing values for the time-dependent variables needed to initialize CLM2.0 from a spun-up state by setting the namelist **FINIDAT**. This file **MUST** have the same atm/ocn resolution as the model run. These variables are initialized to non spun-up values in CLM2.0 if **FINIDAT** is not set.

Finally, the only CLM2.0 namelist variable that is specific to csm mode is **CSM_DOFLXAVE**. If this variable is set to true, flux averaging is perfromed over the duration that the namelist variable **IRAD** is set to.

4 History File Fields

In the following table we list the different fields that are currently output to CLM2.0 history files. The dimensions of each field may include time (days ince the beginning of the simulation, lat (number of latitude points, e.g., lat = 64 for a T42 simulation), lon (number of longitude points, e.g., lon = 128 for a T42 simulation), or levsoi (number of soil layers, levsoi = 10). If a field's default state is inactive, it will not appear on a history file unless its state is modified via the namelist variable **FLDADD**.

CLM2.0 History file fields	
Soil time-invariant properties	
WATSAT	
description	saturated soil water content (porosity)
units	mm3/mm3
dimensions	time,levsoi,lat,lon
default averaging	constant (lakes, wetlands, glaciers excluded)
default active/inactive	active
SUCSAT	
description	saturated soil matric potential
units	mm
dimensions	time,levsoi,lat,lon
default averaging	constant (lakes, wetlands, glaciers excluded)
default active/inactive	active
BSW	
description	slope of soil water retention curve
units	unitless
dimensions	time,levsoi,lat,lon
default averaging	constant (lakes, wetlands, glaciers excluded)
default active/inactive	active
ZSOI	
description	"node" depth of soil layer (depth of soil layer at which hydraulic/thermal properties are defined)
units	m
dimensions	time,levsoi,lat,lon
default averaging	constant (lakes excluded)
default active/inactive	active
DZSOI	
description	soil layer thickness
units	m
dimensions	time,levsoi,lat,lon
default averaging	constant (lakes excluded)
default active/inactive	active
Temperatures	
TSA	
description	2 m air temperature
units	K
dimensions	time,lat,lon

default averaging	average
default active/inactive	active
TV	
description	vegetation temperature
units	K
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
TG	
description	ground temperature
units	K
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
TSOI	
description	soil temperature
units	K
dimensions	time,levsoi,lat,lon
default averaging	average (lakes excluded)
default active/inactive	active
TLAKE	
description	lake temperature
units	K
dimensions	time,levsoi,lat,lon
default averaging	average (averaging done only for valid (lake) subgrid patches)
default active/inactive	active
TSNOW	
description	snow temperature
units	K
dimensions	time,lat,lon
default averaging	average (horizontal averaging done only for valid subgrid patches (those which have snow, lakes excluded), vertical averaging is done over all valid snow layers)
default active/inactive	active
Surface radiation	
FSA	
description	absorbed solar radiation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FSR	
description	reflected solar radiation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
NDVI	
description	surface normalized difference vegetation index
units	unitless

dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FIRA	
description	net infrared (longwave) radiation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FIRE	
description	emitted infrared (longwave) radiation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
Surface energy fluxes	
FCTR	
description	canopy transpiration
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FCEV	
description	canopy (intercepted) evaporation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FGEV	
description	ground evaporation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FSH	
description	sensible heat
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FGR	
description	heat flux into snow/soil (includes snow melt)
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FSM	
description	snow melt heat flux
units	watt/m2
dimensions	time,lat,lon

default averaging	average
default active/inactive	active
TAUX	
description	zonal surface stress
units	kg/m/s2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
TAUY	
description	meridional surface stress
units	kg/m/s2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
Vegetation phenology	
ELAI	
description	exposed one-sided leaf area index
units	m2/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
ESAI	
description	exposed one-sided stem area index
units	m2/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
Canopy physiology	
RSSUN	
description	sunlit leaf stomatal resistance
units	s/m
dimensions	time,lat,lon
default averaging	minimum (lakes excluded)
default active/inactive	active
RSSHA	
description	shaded leaf stomatal resistance
units	s/m
dimensions	time,lat,lon
default averaging	minimum (lakes excluded)
default active/inactive	active
BTRAN	
description	transpiration beta factor (soil moisture limitation)
units	unitless
dimensions	time,lat,lon
default averaging	average (lakes excluded)
default active/inactive	active
FPSN	
description	photosynthesis
units	unitless
dimensions	time,lat,lon

default averaging	average (lakes excluded)
default active/inactive	active
Hydrology	
H2OSOI	
description	volumetric soil water
units	mm3/mm3
dimensions	time,levsoi,lat,lon
default averaging	average (lakes excluded)
default active/inactive	active
H2OSNO	
description	snow depth (liquid water equivalent)
units	mm
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
H2OCAN	
description	intercepted water
units	mm
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
SOILLIQ	
description	soil liquid water
units	kg/m2
dimensions	time,levsoi,lat,lon
default averaging	average (lakes excluded)
default active/inactive	active
SOILICE	
description	soil ice
units	kg/m2
dimensions	time,levsoi,lat,lon
default averaging	average (lakes excluded)
default active/inactive	active
SNOWLIQ	
description	snow liquid water
units	kg/m2
dimensions	time,lat,lon
default averaging	average (horizontal averaging done only for valid subgrid patches (those which have snow, lakes excluded), vertical averaging done by summing over all valid snow layers)
default active/inactive	active
SNOWICE	
description	snow ice
units	kg/m2
dimensions	time,lat,lon
default averaging	average (horizontal averaging done only for valid subgrid patches (those which have snow, lakes excluded), vertical averaging done by summing over all valid snow layers)
default active/inactive	active
SNOWDP	

description	snow height
units	m
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
SNOWAGE	
description	snow age
units	unitless
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QINFL	
description	infiltration
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QOVER	
description	surface runoff
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QRGWL	
description	surface runoff at glaciers, wetlands, lakes
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QDRAI	
description	sub-surface drainage
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QINTR	
description	interception
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QDRIP	
description	throughfall
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QMELT	
description	snow melt
units	mm/s

dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QSOIL	
description	ground evaporation
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QVEGE	
description	canopy (intercepted) evaporation
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QVEGT	
description	canopy transpiration
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QCHOCNR	
description	RTM river discharge into ocean
units	m3/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active (if **RTM** defined)
QCHANR	
description	RTM river flow (maximum subgrid flow)
units	m3/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active (if **RTM** defined)
Water and energy balance checks	
ERRSOI	
description	soil/lake energy conservation error
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
ERRSEB	
description	surface energy conservation error
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
ERRSEBMX	
description	maximum of surface energy conservation error
units	watt/m2
dimensions	time,lat,lon

default averaging	maximum
default active/inactive	active
ERRSOL	
description	solar radiation conservation error
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
ERRH2O	
description	total water conservation error
units	mm
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
Atmospheric forcing	
RAIN	
description	rain
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
SNOW	
description	snow
units	mm/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
TBOT	
description	atmospheric air temperature
units	K
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
WIND	
description	atmospheric wind velocity magnitude
units	m/s
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
THBOT	
description	atmospheric air potential temperature
units	K
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
QBOT	
description	atmospheric specific humidity
units	kg/kg
dimensions	time,lat,lon
default averaging	average

default active/inactive	active
ZBOT	
description	atmospheric reference height
units	m
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FLDS	
description	incident longwave radiation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
FSDS	
description	incident solar radiation
units	watt/m2
dimensions	time,lat,lon
default averaging	average
default active/inactive	active
Time-invariant grid fields (these are defined in subroutine histcrt.F90 and should not be changed)	
EDGEN	
description	northern edge of surface grid
units	degrees north
dimensions	scalar
default averaging	constant
default active/inactive	active (if **OFFLINE** defined)
EDGEE	
description	eastern edge of surface grid
units	degrees east
dimensions	scalar
default averaging	constant
default active/inactive	active (if **OFFLINE** defined)
EDGES	
description	southern edge of surface grid
units	degrees south
dimensions	scalar
default averaging	constant
default active/inactive	active (if **OFFLINE** defined)
EDGEW	
description	western edge of surface grid
units	degrees west
dimensions	scalar
default averaging	constant
default active/inactive	active (if **OFFLINE** defined)
LON	
description	coordinate longitude
units	degrees east
dimensions	lon
default averaging	constant
default active/inactive	active

LAT	
description	coordinate latitude
units	degrees north
dimensions	lat
default averaging	constant
default active/inactive	active
LEVSOI	
description	coordinate soil levels
units	m
dimensions	levsoi
default averaging	constant
default active/inactive	active
TIME	
description	time
units	days since beginning of run
dimensions	time
default averaging	constant
default active/inactive	active
LONGXY	
description	longitude
units	degrees east
dimensions	lat,lon
default averaging	constant
default active/inactive	active
LATIXY	
description	latitude
units	degrees north
dimensions	lat,lon
default averaging	constant
default active/inactive	active
AREA	
description	grid cell area
units	km2
dimensions	lat,lon
default averaging	constant
default active/inactive	active
LANDFRAC	
description	land fraction
units	unitless
dimensions	lat,lon
default averaging	constant
default active/inactive	active
NUMLON	
description	number of longitudes at each latitude
units	unitless
dimensions	lat
default averaging	constant
default active/inactive	active
LANDMASK	
description	land/ocean mask

units	(0.=ocean and 1.=land)
dimensions	lat,lon
default averaging	constant
default active/inactive	active
Time-information fields (these are defined in sub routine histcrt.F90 and should not be changed)	
MCDATE	
description	current date as 8 digit integer
units	YYYYMMDD
dimensions	time
default averaging	instant
default active/inactive	active
MCSEC	
description	current seconds of current date
units	s
dimensions	time
default averaging	instant
default active/inactive	active
MDCUR	
description	current day (from base day)
units	unitless
dimensions	time
default averaging	instant
default active/inactive	active
NSTEP	
description	time step
units	unitless
dimensions	time
default averaging	instant
default active/inactive	active
TIME_COMMENT	
description	history interval for time slice
units	character
dimensions	time,string_length(80)
default averaging	instant
default active/inactive	active

5 Offline Mode Namelist Examples

The following examples illustrate different namelist options that can be used to run CLM2.0 in offline mode.

5.1 Example 1: offline initial run, one day, global

When CLM2.0 is run in offline mode, the minimum namelist parameters are: **CASEID**, **NSREST**, **NESTEP** or **NELAPSE**, **FSURDAT**, **FPFTCON**, **OFFLINE_ATMDIR**, **START_YMD**, and **DTIME**. This assumes that a surface dataset exists. If **FSURDAT** is blank, then a surface dataset will be generated at run time and additional variables need to be specified to create that surface dataset (see Section 3.2 and Example 5). All other namelist parameters will be set to the default values. The following gives an example of such a namelist.

```
&CLMEXP
CASEID          = 'test01'
NSREST          = 0
NESTEP          = -1
FSURDAT         = '$MODEL_DATADIR/srfdata/offline/surface-data.120x060.013002.nc'
FPFTCON         = '$MODEL_DATADIR/pftdata/pft-physiology'
FRIVINP_RTM     = '$MODEL_DATADIR/rtmdata/rdirc.05'
OFFLINE_ATMDIR  = '$MODEL_DATADIR/NCEPDATA'
START_YMD       = 19971231
DTIME           = 1800
/
```

CASEID = 'test01'
Distinguishes this particular simulation from another. The string in **CASEID** shows up in the restart pointer file name (see Example 2), in the mass store path where history and restart files are placed if mass store is used, and in the names of history, restart, and initial files. In a branch run, the user must specify a new **CASEID**.

NSREST = 0
Requests an initial run as opposed to a restart or a branch run. An initial run does not require the use of an initial input datafile (**FINIDAT**), so if none is provided (as in this example), the model uses the non spun-up initialization provided in the code (see src/main/iniTimeVar.F90).

NESTEP = -1
Specifies the run's ending time to be at the end of day 1. NESTEP overrides any value given to **NELAPSE**, so **NELAPSE** has been omitted in this example.

FSURDAT = '$MODEL_DATADIR/srfdata/surface-data.120x060.013002.nc'
Defines the name of the surface data input file. This file can be created for a global domain or any domain (even a point) at any resolution as described in example 5. Model resolution (i.e. LSMLON, LSMLAT) is set in the job script and should be compatible with the resolution of **FSURDAT**. The path shown in this example contains the environment variable **MODEL_DATADIR**. The definition of environment variables is explained in section 2.1.1. If the file name appeared without a path specifying its exact location, the file would be expected in the executable directory, defined with environment variable **MODEL_EXEDIR**.

FPFTCON = '$MODEL_DATADIR/pftdata/pft-physiology'
Specifies a file with PFT (Plant Functional Type) information. See explanation for **FSURDAT** above to understand the use of **MODEL_DATADIR** or **MODEL_EXEDIR**.

FRIVINP_RTM
Specifies the input file required for the operation of RTM (the River Transport Model of Branstetter et al). By default, RTM will operate at half degree horizontal resolution and will be invoked every 3 hours, where the RTM input fluxes are averaged over the 3 hour period. If the user wants the RTM scheme to be invoked every timestep, **RTM_NSTEPS** should be set to 1. Otherwise, if the user wants the RTM scheme to operate at a different frequency than once every 3 hours, **RTM_NSTEPS** should be set to the desired value. Use of RTM is activated in the job script with the C pre-processor (cpp) directive #define RTM (see section 2.1.2).

OFFLINE_ATMDIR = '$MODEL_DATADIR/NCEPDATA'
Specifies the location of the atmospheric driver data set. Such a data set is required for the model to run in offline mode.

START_YMD = 19971231
Is the base date of the simulation and must be compatible with the atmospheric input data. For example,

start_ymd = 19971231 corresponds will use the atmospheric input file 1997-12.nc. In a restart or branch run, **START_YMD** need not be changed, as long as it refers to a date earlier than the date of restart or branch.

DTIME = 1800
Specifies the simulation's timestep in seconds. In offline mode, the model can handle a timestep of up to 3600 seconds.

History and restart files will be produced in the executable directory and will be written every 24 hours by default. The frequency of history writes can be controlled as shown in examples 3 and 4.

5.2 Example 2: Restart from example 1 and run for one more day

```
&CLMEXP
CASEID          =  'test01'
NSREST          =  1
NELAPSE         =  -1
FSURDAT         =  '$MODEL_DATADIR/srfdata/offline/surface-data.120x060.013002.nc'
FPFTCON         =  '$MODEL_DATADIR/pftdata/pft-physiology'
FRIVINP_RTM     =  '$MODEL_DATADIR/rtmdata/rdirc.05'
OFFLINE_ATMDIR  =  '$MODEL_DATADIR/NCEPDATA'
START_YMD       =  19971231
DTIME           =  1800
/
```

NSREST = 1
Requests a restart run. A restart run finds the name of the appropriate restart file automatically by reading the file, lnd.**CASEID**.rpointer. In this example, the pointer file will be lnd.test.rpointer. Restart runs are meant to be 'seamless,' producing the same output as runs which were not restarted.

NELAPSE = -1
Specifies the run's ending time to be one day after the point of restart. This is equivalent to entering **NESTEP** = -2 instead, since the previous run stopped at the end of day 1.

All other namelist variables remain the same to ensure a 'seamless' restart (for information, see example 1). Also, for a seamless restart, the user should generally execute the code with the same executable used in the initial run (ie, without compiling the code again).

5.3 Example 3: Branch run from example 1 for one day

```
&CLMEXP
CASEID          =  'branch_run'
NSREST          =  3
NELAPSE         =  -1
FSURDAT         =  '$MODEL_DATADIR/srfdata/offline/surface-data.120x060.013002.nc'
FPFTCON         =  '$MODEL_DATADIR/pftdata/pft-physiology'
FRIVINP_RTM     =  '$MODEL_DATADIR/rtmdata/rdirc.05'
OFFLINE_ATMDIR  =  '$MODEL_DATADIR/NCEPDATA'
NREVSN          =  'test01.clm2.r.1998-01-01-00000'
HIST_NHTFRQ     =  -3
START_YMD       =  19971231
DTIME           =  1800
/
```

See example 1 for explanations of namelist variables which remain unchanged.

NSREST = 3
Requests a branch run.

NELAPSE = -1
Specifies the run's ending time to be one day after the point of branching.

NREVSN = 'test.clm2.r.1998-01-01-00000'
Supplies the name of the restart file which will initialize this run. (Note this file can be produced by running example 1 above).

HIST_NHTFRQ = -3
Changes the frequency of history writes to every 3 hours. This is an example of a change which a user may wish to test in a branch run.

The user may branch a run with the same executable used in the initial run (i.e., without compiling the code again), unless branching is used to test changes in the code (for debugging or sensitivity purposes).

5.4 Example 4: Offline CLM2 initial run, one month, auxiliary history files

This example covers the addition of fields to a history file, the addition of an auxiliary history file, and the change of field type in a history file. A variety of other namelist options are discussed here, too.

```
&CLMEXP
CASEID          = 'rtm_run'
NSREST          = 0
NESTEP          = -31
FINIDAT         = 'test01.clm2.i.1998-01-01-00000.nc'
FSURDAT         = '$MODEL_DATADIR/srfdata/offline/surface-data.120x060.013002.nc'
FPFTCON         = '$MODEL_DATADIR/pftdata/pft-physiology'
FRIVINP_RTM     = '$MODEL_DATADIR/rtmdata/rdirc.05'
OFFLINE_ATMDIR  = '$MODEL_DATADIR/NCEPDATA'
HIST_DOV2XY     = .true.,.true.
HIST_NHTFRQ     = -12,0
HIST_MFILT      = 4,1
HIST_FLDADD     = 'TLAKE','TSNOW'
HIST_CHNTYP     = 'RSSUN','instant','RSSHA','maximum'
HIST_FLDAUX1    = 'TSA','ELAI'
START_YMD       = 19980101
DTIME           = 1800
WRTDIA          = .true.
/
```

For namelist variables which are repeated, please refer to Examples 1, 2, and 3.

FINIDAT
Specifies an initial file to be used instead of non spun-up initialization. This results in a simulation that starts from a spun-up state. As written, the initial file will be expected in the executable directory. The initial dataset, **test.clm2.i.1998-01-01-00000.nc** can be obtained by running example 1 above.

HIST_DOV2XY = .true.,.true.
History output will appear in gridded 2d xy format (rather than 1d subgrid format) in both the primary and auxiliary history files.

HIST_NHTFRQ = -12,0
History output will be directed to the primary history file every 12 model hours and to the auxiliary history file every month.

HIST_MFILT = 4,1
Each primary history file will contain 4 time slices of output, while each auxiliary history file will contain 1 time slice of output.

HIST_FLDADD = 'TLAKE','TSNOW'
Specifies two fields to be added to the primary history output. This is a temporary equivalent to changing the logical variable 'active' to .true. for these two fields in routine **src/main/histFileMod.F90**.

HIST_CHNTYP = 'RSSUN','instant','RSSHA','maximum'
Changes the history output for these fields to "instantaneous" and to "maximum." This information can also be hardwired in subroutine **histlst**.

HIST_FLDAUX1 = 'TSA','ELAI'
Places these two fields in the first (and only, in this case) set of auxiliary history files. The code is written to allow for up to 2 sets of auxiliary history files. These variables need to be 'active' in subroutine **histlst**.

WRTDIA = .true.
A global average of surface air temperature as diagnostic will appear in the standard output file of the simulation.

5.5 Example 5. Make offline surface dataset at regular grid and run for a day

CLM2.0 can create a surface dataset necessary to run the model at any horizontal resolution. This dataset, specified by the setting of **FSURDAT**, includes information about the land surface (eg, vegetation, soils, inland water). A regular grid surface dataset can be generated for a single gridcell or for several gridcells that comprise a regional domain. To generate a surface dataset for a regional run, the cpp tokens **LSMLON** and **LSMLAT** must be set to the desired resolution (e.g., **LSMLON**=1, **LSMLAT**=1 for a single point simulation) and the variables **MKSRF_OFFLINE_EDGES**, **MKSRF_OFFLINE_EDGEN**, **MKSRF_OFFLINE_EDGEE**, and **MKSRF_OFFLINE_EDGEW** and their values need to be added to the namelist. A surface dataset will be created with the name surface-data.LSMLONXLSMLAT.nc (e.g., for a single point simulation the file name will be surface-data.001x001.nc). CLM2.0 can then be run for the single point or regional domain by following Example 1 where **FSURDAT** is set to the new surface dataset.

The following namelist will result in the generation of a surface dataset on a global regular grid.

```
&CLMEXP
CASEID                    =   'create_surfdat'
NSREST                    =   0
NESTEP                    =   -1
START_YMD                 =   19971231
DTIME                     =   1800
FSURDAT                   = ' '
FRIVINP_RTM               =   '$MODEL_DATADIR/rtmdata/rdirc.05'
FPFTCON                   =   '$MODEL_DATADIR/pftdata/pft-physiology'
OFFLINE_ATMDIR            =   '$MODEL_DATADIR/NCEPDATA'
MKSRF_OFFLINE_FNAVYORO    =   '$MODEL_DATADIR/rawdata/mksrf_navyoro_20min.nc'
MKSRF_FVEGTYP             =   '$MODEL_DATADIR/rawdata/mksrf_pft.nc'
MKSRF_FSOITEX             =   '$MODEL_DATADIR/rawdata/mksrf_soitex.10level.nc'
MKSRF_FSOICOL             =   '$MODEL_DATADIR/rawdata/mksrf_soicol_clm2.nc'
MKSRF_FLANWAT             =   '$MODEL_DATADIR/rawdata/mksrf_lanwat.nc'
MKSRF_FGLACIER            =   '$MODEL_DATADIR/rawdata/mksrf_glacier.nc'
MKSRF_FURBAN              =   '$MODEL_DATADIR/rawdata/mksrf_urban.nc'
MKSRF_FLAI                =   '$MODEL_DATADIR/rawdata/mksrf_lai.nc'
/
```

FSURDAT = ' '
A surface dataset named surface-data.LSMLONXLSMLAT.nc will be created in the model executable directory. LSMLON and LSMLAT are defined in the jobscript (see section 2.1.2).

NESTEP = 1
No model output is produced from this run, other than a new surface data file. The user could alternatively create the surface data file and continue the simulation uninterupted.

MKSRF_OFFLINE_FNAVYORO = '$MODEL_DATADIR/rawdata/mksrf_navyoro_20min.nc'
Points to the orography dataset used to derive the model's land mask in offline mode. The environment variable **$MODEL_DATADIR** is explained in section 2.1.1.

MKSRF_FVEGTYP, **MKSRF_FSOITEX**, **MKSRF_FSOICOL**, **MKSRF_FLANWAT**, **MKSRF_FGLACIER**, **MKSRF_FURBAN**, and **MKSRF_FLAI**
specify the input datasets used to create the surface dataset. The environment variable **$MODEL_DATADIR** is explained in section 2.1.1.

MKSRF_OFFLINE_EDGES, **MKSRF_OFFLINE_EDGEN**, **MKSRF_OFFLINE_EDGEE**, and **MKSRF_OFFLINE_EDGEW**
must be defined when the desired model domain is not global. The units are degrees north for edges and edgen and degrees east for edgee and edgew.

5.6 Example 6. Make offline surface dataset on global gaussian grid and run for a day

Only global surface datasets can be created on a non-regular grid, such as a gaussian grid. To generate a surface dataset for on a gaussian grid, the cpp tokens **LSMLON** and **LSMLAT** must be set to the desired resolution (e.g., **LSMLON**=128, **LSMLAT**=64 for a T42 grid, and **MKSRF_OFFLINE_FGRID** must be set to the appropriate dataset in **$MODEL_DATADIR/rawdata** which specifies the appropriate grid, land mask and land fraction to use when creating the surface datset. A surface dataset with the name, surface-data.LSMLONXLSMLAT.nc, will be created in the executable directory. This dataset may be renamed by the user to be more self-explanatory.

The following namelist will result in the generation of a surface dataset on a global regular grid.

```
&CLMEXP
CASEID              = 'create_surfdat'
NSREST              = 0
NESTEP              = -1
START_YMD           = 19971231
DTIME               = 1800
FSURDAT             = ' '
FRIVINP_RTM         = '$MODEL_DATADIR/rtmdata/rdirc.05'
FPFTCON             = '$MODEL_DATADIR/pftdata/pft-physiology'
OFFLINE_ATMDIR      = '$MODEL_DATADIR/NCEPDATA'
MKSRF_OFFLINE_FGRID = '$MODEL_DATADIR/rawdata/T42_clm2_camfgrid_040802.nc'
MKSRF_FVEGTYP       = '$MODEL_DATADIR/rawdata/mksrf_pft.nc'
MKSRF_FSOITEX       = '$MODEL_DATADIR/rawdata/mksrf_soitex.10level.nc'
MKSRF_FSOICOL       = '$MODEL_DATADIR/rawdata/mksrf_soicol_clm2.nc'
MKSRF_FLANWAT       = '$MODEL_DATADIR/rawdata/mksrf_lanwat.nc'
MKSRF_FGLACIER      = '$MODEL_DATADIR/rawdata/mksrf_glacier.nc'
MKSRF_FURBAN        = '$MODEL_DATADIR/rawdata/mksrf_urban.nc'
MKSRF_FLAI          = '$MODEL_DATADIR/rawdata/mksrf_lai.nc'
/
```

FSURDAT = ' '
A surface dataset named surface-data.LSMLONXLSMLAT.nc will be created in the model executable directory. LSMLON and LSMLAT are defined in the jobscript (see section 2.1.2).

NESTEP = 1
No model output is produced from this run, other than a new surface data file. The user could alternatively create the surface data file and continue the simulation uninterupted.

MKSRF_OFFLINE_FGRID = '$MODEL_DATADIR/rawdata/T42_clm2_camfgrid_040802.nc'
Points to the dataset containing the model grid, land mask and fractional land for the surface dataset.

MKSRF_FVEGTYP, **MKSRF_FSOITEX**, **MKSRF_FSOICOL**, **MKSRF_FLANWAT**, **MKSRF_FGLACIER**, **MKSRF_FURBAN**, and **MKSRF_FLAI**
specify the input datasets used to create the surface dataset. The environment variable **$MODEL_DATADIR** is explained in section 2.1.1.

6 Testing Model Changes

The script test-model.pl in the bld/offline/tests directory runs a suite of basic tests for the CLM2.0 model running in offline mode on a 3x3 model grid. In order to use test-model.pl, the user needs to have at least Perl version 5.004 on their system. This test suite is designed for model testing when the user is modifying model code or simply porting the code to another machine. The test suite provides a check that the basic functionality of the model still works despite the changes the user has introduced.

In this section the usage of test-model.pl for basic acceptance testing is described. The command-line arguments to "test-model.pl" are summarized as well environment variables that are useful to use the script.

The script, test-model.pl, is designed to be run interactively. Although, this script can be run at other labs, testing has only be done on NCAR machines. Consequently, the following discussion will be confined to NCAR platforms. The user should feel free to try the scripts at other labs.

Before running the CLM2.0 test suite, the user needs to set the environment variable, MODEL_DATADIR, specifying the full disk pathname containing the untarred subdirectories **NCEPDATA**, **inidata**, **pftdata**, **rawdata**, **rtmdata** and **srfdata**. At NCAR, CCSM input datasets are stored on a NFS mounted directory,

"/fs/cgd/csm/inputdata/lnd/clm2". Consequently, this is the default value for $MODEL_DATADIR in the testing scripts. If the user has untarred the source code such that the test-model.pl script is in the directory "/home/user/clm2/bld/offline/tests", then the user may invoke the script as follows:

cd /home/user/clm2/bld/offline/tests
test-model.pl

The script, test-model.pl, is designed such that the common settings the user might want to control are easily set by command line arguments. The "-h" option to test-model.pl lists all of the possible command-line options.

-h	Help (this message)
-t	List the tests that are performed
-clean	Clean the old directories out
-nofail	Continue even if errors are found
-r	Continue a previous run of the script at the point it left off at
-s n	Skip to given test n (or range of numbers)
	(example -s 4 start with test no. 4)
	(or -s 2-4 do tests 2 through 4)
-c dir	Compare to another version of the CLM2 model in this directory
	(example -c /home/user/clm2/src)
-l lab	Set the lab you are running at

The "-t" option to "test-model.pl" lists all of the possible tests where there associated numbers. Below is a brief summary of each test.

- Tests 1 and 2. Run three time-steps with DEBUG compiler flags on, with SPMD on and off, and then ensure that answers are identical.
- Tests 2,3 and 4.Perform an initial run followed by a restart using fewer SPMD tasks (if SPMD enabled), and fewer threads. Then do an initial run the same number of timesteps as the restart, and compare answers ensuring they are identical.
- Test 6 If a comparison to a previous code library is requested (via the "-c" option), repeat the last initial run with the previous code library and check if answers are identical.

An important feature of test-model.pl is the ability to compare modified CLM2.0 code to a previous program library. This is useful in order to ensure that the implemented modifications do not change answers if that is what is expected. Using the command line option "-c dir" the user can compare to a previous program library by giving the full path to the root of the library to compare to. For example, if a test library can be found in "/home/user/clm2mod/src" and the unmodified library is located in "/home/user/clm2/src" then "-c" can be used as follows:

cd /home/user/testmod/bld/offline/tests
test-model.pl -c /home/user/clm2/src

When the "-c" option is used, only test 6 is compared to test 5 (see the list of tests above). If this comparison isidentical, the modified model is identified as being bit-for-bit with the control library. This is reported at the end of test-model.pl (and in the log file) as follows:

Many times differences with respect to a control library are intended to be bit-for-bit. As explained above simply by using the "-c" option to test-model.pl can easily identify if two model libraries give identical answers. However, it is more difficult to verify if changes are within machine roundoff. Currently, if non bit-for-bit differences occur, differences between the history files produced by both the control as well as modified libraries must be examined to determine if these differences indicate only roundoff level changes.