Community Sea Ice Model (CSIM) User's Guide
Version 4.0.1

Released with CCSM2.0.1

Julie Schramm

Contents

1 Introduction

1.1 What is CSIM4?

The Community Sea Ice Model (CSIM4) is a component of the Community Climate System Model (CCSM2), a fully-coupled global climate model that provides state-of-the-art computer simulations of the Earth's past, present and future climate states. The sea ice component includes the elastic-viscous-plastic (EVP) dynamics scheme, an ice thickness distribution, energy-conserving thermodynamics, a slab ocean mixed layer model, and the ability to run using prescribed ice concentrations. It is supported on high and low-resolution Greenland Pole grids, which are identical to those used by the Parallel Ocean Program (POP) ocean model. The high resolution version is best suited for simulating present-day and future climate scenarios while the low resolution option is used for paleoclimate simulations and debugging.

In addition to this User's Guide, there is a Scientific Document that describes the model physics and a Reference Guide with a detailed description of the ice model source code. This user's guide describes how to obtain and use CSIM4 and is intended for users who want to run the model with minor changes introduced via existing namelist options. Users interested in modifying the source code should see the Reference Document.

Although it is possible to run this model in a stand alone mode, this option is not discussed here. All configurations presented in this document exchange information with other model components via a flux coupler. An attempt has been made throughout this document to provide the following text convention. Namelist names and variables are `typewritten`. Directory and file names are given in *italic*.

1.1.1 Overview of CSIM4 within CCSM2

The directory structure of the fully coupled CCSM2 is shown in Figure 1. In general, CCSM2 consists of four models (atmosphere, ice, land, ocean) plus a coupler for exchanging information between these components. Each of the four models consists of an "active" version with state-of-the-art physics and a data, version that reads forcing input files. CCSM2 can be configured in a number of ways using various combinations of the components. These configurations are defined by letters. For example, B represents the fully-coupled run that consists of the Community Atmosphere Model (CAM2), the Community Land Model (CLM2), the POP ocean model and the CSIM ice model, all communicating through the coupler. This is the default setup for CCSM2. Definitions of all the model configurations should be in the CCSM2 User's Guide.

There are two supported configurations that include the ice model as the only active component along with the data ocean, a data atmosphere, and the data land model. These setups provide more rapid simulations than the fully coupled model and are also useful for testing the ice model. The directory structure of this configuration is shown in bold in Figure 1.

The first configuration uses a data ocean model called docn5 and a model output data atmosphere called datm5; this setup is called the D configuration. datm5 uses output from previous Community Climate Model (CCM) atmospheric simulations, and does not result in the best sea ice simulation. All forcing fields are read in from a single file, with a separate file for each monthly average. Running the ice model with docn5 will not allow ice growth in open-ocean grid cells, since the ocean mixed layer temperature is fixed at or above the freezing point of sea water. Congelation ice growth will occur with docn5.

The second configuration uses a climatological data atmosphere model called latm5 and the ocean mixed layer within the ice model; this is referred to as the M configuration. latm5 is more flexible than datm5 in that it allows for specification of multiple datasets (observed or reanalysis) for atmospheric forcing. It can also read in combinations of datasets with different temporal resolution, for example, 6 hourly air temperature and monthly averged precipitation. Unlike datm5, latm5 reads in each of the forcing fields from a separate file, which makes it easy to swap in different forcing datasets. One year of NCEP data has been provided with this release of CCSM2.

The recommended configuration for testing ice model physics and examining the ice cover response to external forcing is the M configuration, using latm5 on the fine resolution grid with the slab ocean mixed layer within the ice model.

The source code for the active components, the data models and the coupler are in the */models* directory. CCSM2 and each component has a */doc* directory containing model documentation. The Makefile and the Macros files reside in */models/bld*. All configurations of CCSM2 use the same Makefile. The main scripts for

running the coupled model and the setup scripts for the individual components are in the */scripts/test.a1* directory.

1.1.2 Overview of Running CSIM4 within CCSM2

All configurations are run using two levels of C shell setup scripts. A model run is controlled by a main driver script that sets build and environment variables common to all components. The main script invokes the setup scripts for the coupler, atmosphere, ocean, land and ice components. These setup scripts collect the source code, build the executable, access necessary datasets, set individual component parameter values and generate a namelist file for each component. The main script then executes all of the components simultaneously.

1.1.2.1 What do I need to run CCSM2? Two target architectures are supported: IBM and SGI. The current release will run on Compaq hardware, but will not restart exactly. This will be fixed shortly.

- Operating System: IBM AIX, or SGI IRIX64

- Tools: gunzip, gnumake, Perl5 or greater, Perl/Tk (for GUI)

- Compilers: Fortran90, C

- Permanent disk space

- Temporary disk space

- Libraries: MPI, netCDF

- Input data : 0.26 GB for T31_gx3v4 grid, 0.84 GB for T42_gx1v3 grid, 1.1 GB for T62_gx1v3

- CCSM2.0 source code: 17 MB

1.1.3 IMPORTANT: What's new in CCSM2.0.1?

Changes made to CCSM2.0 were limited so that the CCSM2.0.1 release would give bit-for-bit answers with CCSM2.0 when using all active models on the T42_gx1v3 grid. This is the configuration that produced the 1000 year control run. No changes have been made to the model physics in CCSM2.0.1.

New to CCSM2.0.1 is a paleoclimate configuration that runs on the T31_gx3v4 (coarse resolution) grid. Compared to the gx3 grid not officially released with CCSM2.0, the land mask in gx3v4 has been changed so that Hudson Bay is now open. It is important to note that the ice and snow albedos in this configuration have been reduced by 10% in the northern hemisphere from the values normally used in the sea ice model. These changes have been included as a seperate set of code modifications and setup scripts that is known by the undescriptive case name *test.a2*. Unfortunately, the CCSM2.0 control run has a very similar case name of *test.a1*, so be careful what scripts directory you choose to set up your model simulations.

CCSM2.0.1 also includes new forcing datasets for the data models. datm5 and dlnd5 use new forcing files produced from a coupled run using the latest version of CAM2. A monthly averaged climatology of ice concentration and sea surface temperature is available for use with dice5 and docn5. This ice concentration and sea surface temperature datasets were used to create the datm5 and dlnd5 datasets, so they should be consistent. The sea ice concentration dataset is used by the prescribed ice model.

2 Obtaining the Source Code and Input Datasets

The CSIM4 source code and datasets can be obtained via a web download. The source code should be extracted from the tar file in a small, permanent disk, such as your home directory or a cross-mounted file system. If possible, the data input files should also be extracted on a large, permanent cross-mounted disk. These files are copied to the temporary disk during the build stage.
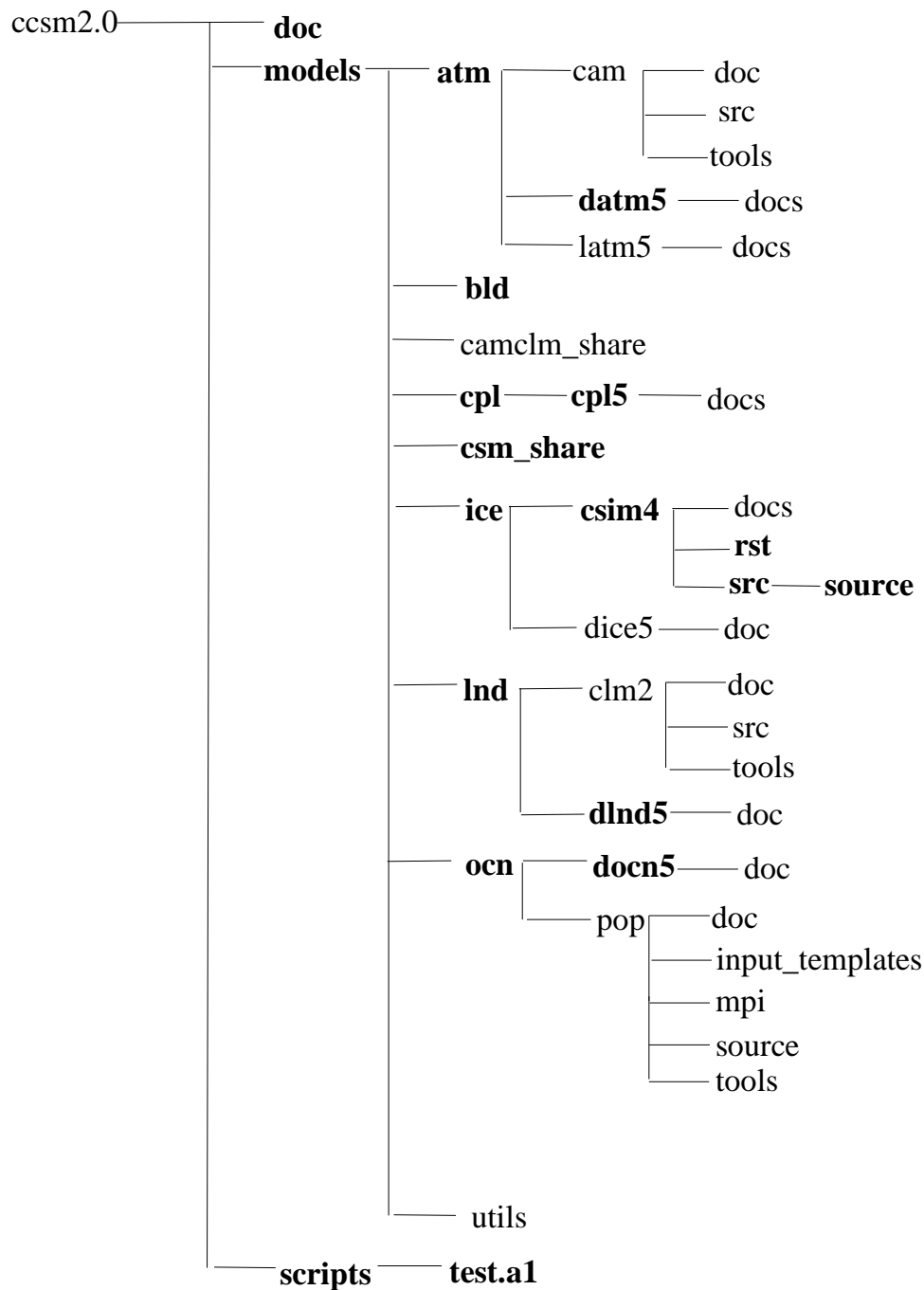
Figure 1: Directory Tree for CCSM2, with the active ice only (D) configuration shown in bold.

2.1 Downloading the Source Code and Input Data

The source code and input datasets are available on the World Wide Web under the URL:
    http://www.ccsm.ucar.edu/models/ccsm2.0.1
The source code comes in the following form:

- *ccsm2.0.1.tar.gz*

To uncompress and untar this file, use the following Unix gunzip and tar commands:
gunzip -c ccsm2.0.1.tar.gz | tar -xf -

The input datasets for CCSM2 are available at the same URL. If the coarse resolution grid is desired, the *T31/gx3v4 input datasets tar file* should be downloaded; *T42/gx1v3 input datasets tar file* should be downloaded for the fine resolution grid. If latm5 is used, the *T62 input datasets tar file* is required in addition to *T42/gx1v3*.

3 Quick Start Guide

It is assumed that the user has obtained and extracted the source code and input data as described in Section 2. This section is intended to get CCSM2 running "out of the box" with a minimal amount of information. It is aimed towards users that are familiar with the script files, the source code and the directory structure and is planning to run the model on one of the supported platforms: IBM, SGI or Compaq. A more detailed description of creating and running the executables is given in Section 4 and in the CCSM2 User's Guide. The default configuration is a 5 day, fully-coupled startup run on the high resolution (T42_gx1v3) grid. Running this configuration verifies that all the input data is in the correct place and the environment variables are set correctly before any further changes are made to the scripts or the source code. This will also provide benchmark output.

3.1 Graphical User Interface

The easiest way to set up the scripts for a particular configuration is to use the graphical user interface (GUI) that comes with the CCSM2 source code. The GUI is a Perl/Tk script that resides in */scripts/gui_run* and is very easy to use. It has been tested on Sun computers and generates scripts for any of the supported CCSM2 model configurations. To start the GUI, type ccsm_gui.pl. A menu will come up with choices for platforms and grid resolution. Boxes are available to fill in source code base directory, case name, input data directory and executable directory. The scripts generated from the GUI will be put in the directory *$CSMROOT/scripts/CASENAME*. This directory will contain the set of scripts from */scripts/test.a1* with the appropriate modifications for the desired configuration. The file called *CASENAME.run* is the script for a one month startup run. The configurations most relevant for the ice model are B, D, M and F and are discussed next.

3.2 Running the Fully Coupled (B) Configuration at NCAR

The B configuration is the fully coupled model with active atmosphere, ice, land and ocean components communicating through the flux coupler. This is the default configuration for CCSM2 and will require minimal changes to the scripts. To run CCSM2 off site, see section 3.6. If you are running the B configuration on the IBM at NCAR, the following lines will need to be changed in the batch system information at the top of the main setup script:

```
# @ wall_clock_limit = 3800
# @ class = csl_reg
# @ account_no = 00000000
```

Modifications are also needed in the main driver script to set the directories for the source code, input data, and executables. The following is a list of the environment variables that will need to be changed by the user:

```
setenv CASE        test.a1                    # case name
setenv CASESTR "fully coupled $GRID test"     # short descriptive text string
setenv CSMROOT  /fs/cgd/home0/$LOGNAME/ccsm2  # root directory of source
setenv CSMDATA  /fs/cgd/csm/inputdata         # base dir for input data
setenv EXEROOT  /ptmp/$LOGNAME/$CASE          # run model here
```

$CASE should be an identifier for the run. It should be kept as short as possible, since it is used in the output file names. Fewer than 8-10 characters is desirable. $CSMROOT is the top directory of the source code. $LOGNAME is set at login and does not need to be set in the main script file. $CSMDATA points to the location of the input data sets. $EXEROOT is typically a large temporary disk where the executable files, input data sets will reside during execution. Information output by the model will also be written to $EXEROOT. The main script is called *test.a1.run* and is in the */scripts/test.a1* directory. To execute a startup run, simply submit *test.a1.run*.

The model output will be in subdirectories in the $EXEROOT directory. An *ice.log* file will be written in *$EXEROOT/ice* that contains a copy of the ice setup script, and information about the compilation and model run. A 150 year simulation with the fully coupled model on the T42_gx1v3 grid will take about one month on 144 processors.

3.3   Running the Active Ice Only (D) Configuration at NCAR

The D configuration consists of the active ice model CSIM4 and data atmosphere, ocean and land components. This configuration runs quickly and is used for testing the software engineering aspects of the model. This configuration will not result in the best sea ice simulation, since docn5 does not allow ice growth in open ocean regions or leads, and the data read in by datm5 is from previous atmospheric model (CCM) simulations. If you are not using the GUI to set up the scripts, you will need to change the following variables in the main script in *scripts/test.a1.run*:

```
# @ task_geometry = {(0,1,10)(2,3,4,5)(6,7,8,9)(11)}

setenv CASESTR 'ccsm2 T42 gx1v3 datm dlnd ice docn cpl' # short descriptive text string

setenv CSMROOT  /fs/cgd/home0/$LOGNAME/ccsm2           # root directory of source
setenv CSMDATA  /fs/cgd/csm/inputdata                  # base dir for input data
setenv EXEROOT  /ptmp/$LOGNAME/$CASE                   # run model here

set SETUPS = (  datm dlnd  ice docn  cpl )             # setup script name

set NTASKS = (     1    1    8    1    1 ) # use NTASK = 1 for data model
set NTHRDS = (     1    1    1    1    4 ) # use NTHRD = 1 for data model
```

If you execute *test.a1.run*, a 5 day startup run will begin. Changes to the batch system information discussed in section 3.2 may also be necessary. Modifications to a variety of input variables are discussed in section 4.6. Details on changing the number of processors are given in section 4.6.2. Changing the grid resolution is described in section 4.6.1.

3.4   Running the Ice Model with the latm5 Atmosphere (M Configuration)

This setup will result more realistic sea ice simulations. One year of NCEP forcing for latm5 is included with this CCSM2 release. This setup is the D configuration discussed in section 3.3, with the latm5 data atmosphere model and the ocean mixed layer model within the ice model. The ocean mixed layer is a simple slab model which computes an ocean surface temperature and allows for ice formation due to ocean supercooling. More information on the formulation of this model can be found in the Scientific Document. latm5 runs on the T62 grid; the GUI can be used to set up the scripts for this configuration. If you are not using the GUI, first make the changes listed in section 3.3 for the D configuration. The following additional changes should be made in the main script:

```
setenv GRID        T62_gx1v3                    # T42_gx1v3 or T31_gx3v4
set SETUPS = ( latm  dlnd  ice docn  cpl )  # setup script name
```

For multi-year simulations, set

```
# @ wall_clock_limit = 21600
```

In *ice.setup.csh*, set

```
set OCEANMIXED_ICE      = .true.
```

In *dlnd.setup.csh*, set

```
data_year0 =  16
data_nyear =  16
```

so that land data from the T62 grid will be used. When these values are set to 5, data from the T42 grid is used.

In the *latm.setup.csh* namelist, `data_nyear` determines how many years of forcing data will be used; the default value is one. In this case, one year of data will be used repeatedly. `ncpl` specifies how many times per day latm5 exchanges information with the coupler. There are other parameters in the latm namelist that may be of interest. `data_oyear` is the coupler year matched to `data_year0`. This configuration runs at about 15 minutes wall clock time per model month using 16 processors for the ice model, and about 30 minutes wall clock time per model month using 8 processors. Both simulations were done on the T62_gx1v3 grid. This setup does not run as efficiently as the B or F configurations, since the buffer sent to and received from the coupler is not packed as efficiently.

3.5   Running the Prescribed Ice (F) Configuration at NCAR

The F configuration is the prescribed ice model with an active atmosphere and land model and the data ocean. In the F run, the ice concentration is prescribed. This is used as a means of tuning the atmosphere model or doing atmosphere-only simulations. The following changes should be made to the main script to run the F configuration:

```
# @ wall_clock_limit = 21600
# @ task_geometry = {(0)(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)(11,12,13,14)(15,16,17,18)(19
,20,21,22)(23,24,25,26)(27)(28)}
set SETUPS = (   atm  lnd  ice docn  cpl )
set NTASKS = (    8    3   16    1    1 )  # use NTASK = 1 for data model
set NTHRDS = (    4    4    1    1    4 )  # use NTHRD = 1 for data model
```

In *ice.setup.csh*, the following changes need to be made:

```
set PRESCRIBED_ICE      = .true.
set PRESCRIBED_ICE_CLIM = .true.
```

Information about further changes to the ice setup script and the necessary input data files can be found in section 4.6.6. This configuration runs at the rate of about 31 minutes wall clock time per model month on the T42_gx1v3 grid with the ice model using 16 processors.

3.6   Running CSIM4 Off Site

There is a directory called *scripts/tools* that contains some supporting scripts. A description of all the scripts in this directory can be found in the CCSM2 User's Guide. Two sets of these scripts are of interest for running the model off site. The first set is *modules.*.**. These are machine specific setup scripts that are called by the main script. The second set of scripts is *test.a1.mods.**. These files contain notes about how to change the main setup script to get it running on other machines. For example, to run CCSM2 on nirvana at Los Alamos National Laboratory (LANL) see *test.a1.mods.nirvana*. The following job control language will need to be added to the top of the main script:

```
#BSUB -L /bin/tcsh
#BSUB -n 36                # Specify threads/processes
#BSUB -R "span[ptile=36]"  # all processors on 1 machine
#BSUB -q csmq              # Specify queue
#BSUB -J test.a1           # Specify the job name
```

The following environment variables should be changed in the main script:

```
setenv CSMROOT  /users/$LOGNAME/ccsm2            # root directory of source
setenv CSMDATA  /n/n07/scratch/$LOGNAME/inputdata # base dir for input data
setenv EXEROOT  /n/n07/scratch/$LOGNAME/$CASE     # run model here
setenv ARCROOT  /n/n07/scratch/$LOGNAME/archive/$CASE  # archive root direct

set NTASKS = (    1    1    4    16    1 ) # use NTASK = 1 for data model use
NTASK = 1 for data model
set NTHRDS = (    8    2    1    1    2 )  # use NTHRD = 1 for data model use
NTHRD = 1 for data model

unsetenv _DSM_VERBOSE         ; setenv _DSM_PLACEMENT ROUND_ROBIN
```

It is possible to use the GUI to set up the model to run on an SGI (select SGI under the Select Platform options). More details on the GUI are in section 3.1.

4   Using CSIM4

4.1   How to Build and Run CSIM4 within CCSM2

Building and executing any configuration of CCSM2 is carried out in the scripts. This section describes the build environment and the scripts. It is assumed that the user has done a 5 day startup run using a configuration described in Section 3.

4.1.1   The Build Environment

All components of CCSM2 use the same Makefile and make environment. These files are located in the *models/bld* directory. There is a single makefile for all models and all platforms. However, there is a *Macros.** file for each platform. *Macros.AIX* is for the IBM, *Macros.IRIX64* is for the SGI, and *Macros.OSF1* is for the Compaq. These files contain information for the preprocessor and the compiler. CSIM4 has a set of options that is different from the default values at the top of the file. These are after the line `ifeq ($(MODEL),csim)` in the *Macros.** files and are described below.

4.1.1.1   CSIM4 Preprocessor Flags Preprocessor flags are activated in the form `-Doption` in the *Macros.** files. The flags specific to the ice model are

```
CPPDEFS :=  $(CPPDEFS) -Dcoupled -DNPROC_X=$(NX) -DNPROC_Y=$(NY) -D_MPI
```

The option `-Dcoupled` is set to activate the coupling interface. This will include the source code in *ice_coupling.F*, for example. For uncoupled runs (not supported), it can be set to anything else (i.e. uncoupled) or removed. If the model is run uncoupled or a coupler other than the CCSM2 coupler is used, there is a flag called `-Dfcd_coupled` that will keep the fluxes from being divided by the ice area. In coupled runs, the CCSM2 coupler multiplies the fluxes by the ice area, so they are divided by the ice area in CSIM4 to get the correct fluxes.

The options `-DNPROC_X=$(NX)` and `-DNPROC_Y=$(NY)` set the number of processors used in each grid direction. These values are calculated in *ice.setup.csh* using $NTASK from the main setup script. NX and NY must divide evenly into the grid. This is used only for MPI grid decomposition. The values of NX and

9

NY are not used for OPENMP (not supported) and single processor runs. If NX or NY do not divide evenly into the grid, the model setup will exit from the *ice.setup.csh* script and print an error message to the *ice.log** file.

The flag -D_MPI sets up the message passing interface. This must be set for runs using a parallel environment. To get a better idea of what code is included or excluded at compile time, grep for "ifdef" and "ifndef" in the code. or look at the **.f* files in the */obj* directory if they are not removed by gmake.

4.1.1.2 CSIM4 Compiler Options CSIM4 uses the CCSM2 default compiler options on the SGI and Compaq platforms. On the IBM, the following options are set in *Macros.AIX*:

```
-c -O3 -qstrict -qrealsize=8 -qarch=pwr3 -qtune=auto -I${INCROOT}
```

All but the last option are standard Fortran compiler options. The last flag includes the Earth System Modeling Framework (ESMF) library, which is not currently used by the ice model.

4.1.2 Creating and Running the Executable

It is assumed that the user has obtained and extracted the source code and input data as described in Section 2 and that the appropriate changes have been made to the main setup script as described in section 3. The executable file is created and executed by simply submitting the main script. Below is a brief description of the data model and coupler setup scripts, and a more complete description of the ice model script. A detailed description of the other scripts can be found in the CCSM2 User's Guide.

4.1.3 Setup Scripts

The purpose of the main driver script is to set environment variables that are available to the component model scripts, execute the setup scripts for each component in the desired configuration, and then execute all components simultaneously. Each component model is prepared for execution via a separate script. These scripts create directories, position input data, create a file with the namelist input, and build an executable. This section describes the scripts necessary to setup the D configuration: the active ice component with data models for the ocean, atmosphere, and land.

4.1.3.1 Main Driver Script The main driver script for CCSM2 is *scripts/test.a1/test.a1.run*. The script is set up for the default B configuration. To run CCSM2 in this configuration, see section 3. A complete description of this script is available in the CCSM2 User's Guide. Several changes will have to be made to this file to set up the D configuration. These changes are discussed in section 3.3.

4.1.3.2 Ice Setup Script The purpose of the ice setup script, *ice.setup.csh*, is to build an executable version of the ice model, document what source code and data files are being used in the *ice.log.$LID* file, and gather or create any necessary input data files. $LID is a time stamp set in the main script. This file is executed from the main driver script. The environment variables shown in Table 1 are set in the main script and used in the ice setup script. These variables are described in the CCSM2 User's Guide.

The environment variables set locally in the ice setup script are listed in Table 2. $HSTDIR, $RSTDIR and $INIDIR are the directories in $EXEDIR where the history, restart, and initial condition files are output, respectively. These follow the directory structure set in the CCSM2 output filename requirements. $ICE-DATA is the directory under $CSMDATA where the ice input data is located. The ice model input templates are located in $ICESRC. These templates are fortran modules that contain information on grid dimensions, number of ice thickness categories and vertical resolution in the ice categories. The grid is determined in the main script and the resolution $RES is set in *ice.setup.csh*: 100x116 for the gx3v4 grid and 320x384 for gx1v3. Depending on $RES and the number of ice thickness categories $NCAT set in *ice.setup.csh*, the appropriate input template *ice_model_size.${RES}x${NCAT}* will be copied into the ice model source code directory and renamed *ice_model_size.F*. Files exist for the following configurations:

ice_model_size.F.100x116x1
ice_model_size.F.100x116x3

Table 1: Environment Variables in Main Script Used in Ice Setup Script

Symbol	Description	Default Value	Other Values
ARCH	system architecture m	ncar	lanl
BASEDATE	initial start date (YYYY-MM-DD)	0001-01-01	
CASE	case name	test.a1	
CSMBLD	makefile and Macros directory	$CSMROOT/models/bld	
CSMCODE	source code base directory	$CSMROOT/models	
CSMDATA	input data base directory	/fs/cgd/csm/inputdata	
CSMSHR	directory for shared code	$CSMROOT/models/csm_share	
EXEDIR	ice model executable directory	$EXEROOT/ice	
GRID	atm and ocean grid	T42_gx1v3	T31_gx3v4
ICE_GRID	ice model grid	gx1v3	gx3v4
MODEL	name of component model	ice	
MSSDIR	MSS directory path name	null:/dev/nul	
MSSPWD	MSS file write password	$LOGNAME	
MSSRPD	MSS file retention period in days	0	
NTASK	number of tasks for ice model	16	4, 8
NTHRD	number of threads for ice model	1	
OBJDIR	component model code is built here	$OBJROOT/ice/obj	
REFCASE	input data case name for branch run	test.a1	
REFDATE	start date for branch run	0001-01-06	
RUNTYPE	run type	startup	continue
SCRIPTS	directory for run scripts	$CSMROOT/scripts/$CASE	

Table 2: Environment Variables Set in the Ice Setup Script

Symbol	Description
HSTDIR	directory where history files are written
RSTDIR	directory where restart files are written
INIDIR	directory where initial condition files are written
ICEDATA	subdirectory where ice model input data collection is located
ICESRC	directory where ice model input templates are located
RES	grid dimensions used to select model resolution
NCAT	number of thickness categories in the ice thickness distribution
NX	number of processors assigned in the x direction, used for MPI grid decomposition
NY	number of processors assigned in the y direction, used for MPI grid decomposition
NLAT	number of latitudes in grid resolution
NLON	number of longitudes in grid resolution
OCEANMIXED_ICE	logical variable used to implement ocean mixed layer model
OML_ICE_SST_INIT	logical variable, if true initialize ocean mixed layer temperature from within ice model
PICE_DATA	datafile containing forcing for prescribed ice model
PRESCRIBED_ICE	logical variable used to implement prescribed ice model
PRESCRIBED_ICE_CLIM	logical variable, if true, climatological data is used with the prescribed ice model
RESTART	logical variable used to initialize model from a restart file
RSTFILE	name of restart file

ice_model_size.F.100x116x5
ice_model_size.F.100x116x10
ice_model_size.F.320x384x1
ice_model_size.F.320x384x3
ice_model_size.F.320x384x5
ice_model_size.F.320x384x10

The total number of processors \$NTASK allocated to the ice model is set in the main script. In *ice.setup.csh*, this number is divided into the number of processors in the x and y directions, \$NX and \$NY. The grid decomposition has been fixed at NY=1 for fewer then 8 processors, NY=2 for 8 or more processors, and NX=NTASK/NY. Currently, NX and NY MUST divide evenly into the grid. There are checks for this in the ice setup script and in the ice source code; the model will stop if these criteria are not met. \$NLAT and \$NLON are used for this purpose.

The environment variable \$OCEANMIXED_ICE is set to `.true.` to implement the slab ocean mixed layer within the ice model. Details on this option and how to set \$OML_ICE_SST_INIT are described in section 4.6.7.

Similarly, the environment variable \$PRESCRIBED_ICE is set to `.true.` to activate the prescribed ice option within the ice model. This option is meant to be run with active atmosphere and land components and a data ocean, and will require changes to the main setup script. Details on setting up the prescribed ice model and a description of \$PRESCRIBED_ICE_CLIM can be found in sections 3.5 and 4.6.6.

The sea ice model contains two namelists. The variables for both lists are set in *ice.setup.csh* and are written to the file *ice_in* in *\$EXEDIR/ice* when the ice setup script is executed. Changes to the namelists must be made in the setup script, not in the *ice_in* file. The ice model reads his file at runtime. The first namelist is called *ice_nml* and is defined in *ice_init.F*. The variables are listed in Tables 3-10. The namelist options are discussed in section 4.3.1.

The second namelist is called *icefields_nml* and is defined in *ice_history.F*. It contains a list of logical

variables that correspond to ice fields that can be written to the history file. By default, all these variables are set to ".true.", so leaving the namelist blank will result in all variables being written to the history file. The available fields are listed in Table 12. Changing the content of the history files via the namelist is discussed in section 4.8.3.3.

4.1.3.3 Data atm Setup Script There are two data atmosphere models. The first is datm5 and uses the setup script *datm.setup.csh*. This data model reads in netCDF output files from a previous CAM atmospheric simulation and interpolates the data in time. The filename for the T31 forcing is *f20.007.cam2.h1.0006-01-01-00000.nc* and is *f20.007.cam2.h1.0006-01-01-00000.nc* for the T42 grid. These filenames are already set in the datm setup script as are the namelist parameters for the years of data that are read in:

```
data_year0 =  0006
data_nyear =  1
data_oyear =  1
```

data_year0 specifies the first year in the input atmosphere data file sequence. data_nyear specifies the number of years in the input atmosphere data file sequence. data_oyear is the simulated year that corresponds to data_year0. Note that only one year of CAM data is provided with this release of CCSM2. Changes to the datm script to read in the appropriate file are made automatically when $GRID is changed in the main setup script.

To modify the solar forcing, a namelist parameter can be added

```
 flux_swfact=  1.3
```

All CCM shortwave fields will be multiplied by this factor. The code for this is also in *models/atm/datm5/data_in_nc.F90*.

The second data atmosphere model is called latm5 and uses the setup script *latm.setup.csh*. This data model can read in NCEP forcing or a combination of any datasets. One year of NCEP data has been included with this release of CCSM2 and consists of air density, specific humidity, wind speed, and air temperature at 10 meters, precipitation, downwelling longwave, and downwelling and upwelling shortwave. To use latm5, the grid for the atmosphere must be set to T62 in the main setup script. This grid configuration is supported only for latm5 runs; other modifications will have to be made in the main script. See section 3.4 and the data model User's Guide for more information.

4.1.3.4 Data lnd Setup Script The namelist parameters in this setup script are similar to those in *datm.setup.csh*. See section 4.1.3.3 for comments on data_dir, data_year0, data_nyear, and data_oyear. The forcong filename is *f20.007.cpl5.ha.0006-xx.nc* for the T31 grid, *f20.006.cpl5.ha.0006-xx.nc* for the T42 grid, and *lsmh_0016.nc* for the T62 grid. Changes to the dlnd script to read the appropriate file are made automatically when $GRID is changed in the main setup script.

See the data model User's Guide for more information.

4.1.3.5 Data ocn Setup Script The docn.setup.csh script is also set up to read in the new Hurrell sea surface temperature dataset. The file name is *sst_clim_hurrell_1x1_020724.nc*, and the docn model will interpolate the data from the 1x1 grid to the ocean grid set in the main script. A comment should be made about when docn is set in $SETUPS in the main script and $OCEANMIXED_ICE is set to .true. in the ice setup script. Data is read in by docn5 and is received by the ice model from the coupler. This information will be written over by the values calculated in the slab ocean model within the ice model. See the data model User's Guide for more information

4.1.3.6 Coupler Setup Script The parameters that determine when a simulation will stop are set in the coupler namelist. The preset values are

```
stop_option = 'ndays'
stop_n      = 5
```

This will stop the simulation after advancing `stop_n` days; `stop_n` is used only when `stop_option = 'ndays'`. To run the model for a year, set

```
stop_n      = 365
```

To stop the simulation when a particular date is reached, set

```
stop_option = 'date'
stop_date   = 19800801
```

Other options for `stop_option` are

```
stop_option = 'newdecade' stop at the start of the next decade
stop_option = 'newyear'   stop on the next January 1st
stop_option = 'newmonth'  stop on the 1st of the next month
```

If `stop_option` is set to any of these other options, no date or namelist stop information is needed.

The frequency at which restart files are written by the coupler and other components is determined by the parameter `rest_freq`. The coupler will send a message to the other components signaling that it is time to write a restart file. The ice model contains its own flag to signal when to write a restart file: `dumpfreq` in the ice model namelist. The ice model will write out restart files if `rest_freq` is true or if `dumpfreq` is true. The preset value for the coupler flag is
`rest_freq = 'monthly'`
Other options for this parameter are
`rest_freq = 'yearly'`
`rest_freq = 'ndays'`
To create restart files every three days, set

```
rest_freq   = 'ndays'
rest_n      = 3
```

`rest_n` is only used if `rest_freq = 'ndays'`. See the coupler User's Guide for more information on these and other parameters.

4.2 Multiple Resubmissions

Multiple runs can be done with a single submission of the main script. To do this, two scripts are necessary. Set

```
setenv RUNTYPE     startup
```

in a script called */scripts/$CASE/$CASE.run*, where $CASE is set in this script. This is the default setup. Make a copy of this file and name it */scripts/$CASE/$CASE.run0*. In */scripts/$CASE/$CASE.run*, set

```
setenv RUNTYPE     continue
```

The number of resubmissions can be entered in the file *RESUBMIT*. To start the run, submit *$CASE.run0*. When this is completed, *$CASE.run* will be submitted over and over by RESUBMIT, until the count reaches zero. The restart pointer files (see section 4.8.2.1are updated automatically, so no files need to be modified between submissions as long as the $RUNTYPE is 'continue'.

If you have used the GUI to set up the scripts, *$CASE.run.startup* will be the script to submit. *$CASE.run* will already have `RUNTYPE` set to `continue` and will be submitted when *$CASE.run.startup* is complete.

4.3 Model Input Variables

4.3.1 Namelist Variables

The ice model setup script contains two namelists. The first, named `ice_in` contains variables that control the model setup, and the second is `icefields_nml`, and controls the fields that are written to the history

files. The latter is described in Section 4.8.3.3. The ice setup script reads the input namelists at runtime, and writes the namelist information to the file *ice_in* in the directory where the model executable is located. Therefore, the namelist will be updated even if the ice model is not recompiled. The variables are set to default values in *ice_init.F*. If they are not set in the namelist *ice_in* in *ice.setup.csh*, they will assume the default values listed in Tables 3-10 and 12, which list all available namelist parameters. Only a few of these variables are required to be set in the namelist; these values are noted in the paragraphs below. An example of the default namelist is shown in Section 4.3.2.1.

Table 3: Namelist Variables for Time Management

Varible	Type	Default Value	Description
runid	Character	'unknown_runid'	Text used in netCDF global attributes
runtype	Character	'unknown_runtype'	Determines if BASEDATE is received from coupler or restart file
istep0	Integer	0	Step counter, number of steps taken in previous integration
npt	Integer	99999	Total number of timesteps in a model run, model stops when istep exceeds npt (not used in coupled runs)
dt	Double	3600.	Timestep in seconds
ndte	Integer	120	Number of subcycles per timestep in ice dynamics

4.3.1.1 Namelist Variables for Time Management The time management namelist options are shown in Table 3. `runid` is a character string that contains descriptive information gathered from the main setup script. This information is written to the global attributes in the history files. `runtype` is determined from the value of $RUNTYPE set in the main setup script. The options for this are discussed in section 4.5. `istep0` is the number of steps taken in a previous integration. It is written to the restart file but is not used to initialize the time for a coupled run. The value `npt` is the upper limit on the number of steps allowed in a model run and is not used in a coupled run, since the point at which to stop integration is determined by the coupler.

`dt` is the timestep in seconds for the ice model thermodynamics and transport. The thermodynamics component is stable but not necessarily accurate for any value of the timestep. The value chosen for `dt` depends on the stability of the transport and the grid resolution. A conservative estimate of `dt` for the transport using the MPDATA advection scheme is

$$\Delta t < \frac{min(\Delta x, \Delta y)}{4max(u, v)} \tag{1}$$

Maximum values for `dt` for the two standard CCSM2 POP grids, assuming $max(u, v) = 0.5 m/s$, are shown in Table 4.

The ice model dynamics is subcycled `ndte` times per timestep so that the elastic waves are damped before the next timestep. The subcycling timestep is calculated as `dte = dt/ndte` and must be sufficiently smaller than the damping timescale `T`, which needs to be sufficiently shorter than `dt`

$$dte < T < dt \tag{2}$$

Table 4: Maximum values for ice model timestep `dt`

Grid	$min(\Delta x, \Delta y)$	$max\Delta t$
gx3v4	28845.9 m	4.0 hr
gx1v3	8558.2 m	1.2 hr

Table 5: Namelist Variables for Writing Output

Varible	Type	Default	Description
`diagfreq`	Integer	24	Frequency of diagnostics written (min, max, hemispheric sums) to standard output 24 => writes once every 24 timesteps 1 => diagnostics written each timestep 0 => no diagnostics written
`histfreq`	Character	'm'	Frequency of output written to history file 'D' or 'd' writes daily data 'W' or 'w' writes weekly data 'M' or 'm' writes monthly data 'Y' or 'y' writes yearly data '1' writes every timestep '0' no history data is written
`hist_avg`	Logical	.true.	If true, averaged history information is written out at a frequency determined by histfreq. If false, instantaneous values rather than time-averages are written.
`dumpfreq`	Character	'y'	Frequency of restart data written to file 'D' or 'd' writes restart files daily 'M' or 'm' writes restart files monthly 'Y' or 'y' writes restart files yearly '0' no restart data is written
`print_points`	Logical	.false.	If true, point data is printed.

This relationship is discussed in Hunke (2001); also see Hunke and Lipscomb (2002), section 4.4. The best ratio for [dte : T : dt] is [1 : 40 : 120]. Typical combinations of `dt` and `ndte` are (3600., 120), (7200., 240) (10800., 120).

4.3.1.2 Namelist Variables for Writing Output   The namelist variables that control the frequency of the model diagnostics, netCDF history files, and binary restart files are shown in Table 5. By default, diagnostics are written out once every 24 timesteps to the ascii file *ice.log.$LID* (see section 4.8.1). $LID is a time stamp that is set in the main script.

`histfreq` controls the output frequency of the netCDF history files; writing monthly averages is the default. The content of the history files is described in section 4.8.3. The value of `hist_avg` determines if instantaneous or averaged variables are written at the frequency set by `histfreq`. If `histfreq` is set to '1' for instantaneous output, `hist_avg` is set to `.false.` within the source code to avoid conflicts. `dumpfreq` controls the output frequency of the binary restart files; writing restart files yearly is the default.

If `print_points` is `.true.`, ascii files are created for the locations shown in Table 6; data is written out every timestep. Additional information is written if the ocean mixed layer model within the ice model is operative. Filenames are of the form *pnt.$BASEDATE.01* are created in the directory *$EXEROOT/ice.*

4.3.1.3 Namelist Variables for Model Physics   The namelist variables for the ice model physics are listed in Table 7. `restart` is always true for coupled runs since all run types begin by reading in a binary restart file. See section 4.5 for a description of the run types and sections 4.5.1.2 and 4.5.1.1 about using restart files and model generated data as initial conditions. `kcolumn` is a flag that will run the model as a single column if is set to 1. This option has not been thoroughly tested and is not supported.

Table 6: Location of Point Data

Location	Latitude	Longitude
SHEBA Camp (Oct 97)	75	-145
SHEBA Camp (July 98)	78	-167
SHEBA Camp (Oct 98)	80	-165
Icelandic Sea	70	-15
Weddell Sea	-65	-40
Ross Sea	-74	-170
East Antarctica	-55	90

Table 7: Namelist Variables for Model Physics

Varible Name	Type	Default Value	Description
restart	Logical	.false.	If true, model is initialized using a restart file, if false, model is initialized using initial conditions in *ice_init.F*.
kcolumn	Integer	0	Column model flag. 0 = off 1 = column model (not tested or supported)
kitd	Integer	1	Determines ITD conversion 0 = delta scheme 1 = linear remapping
kdyn	Integer	1	Determines ice dynamics 0 = No ice dynamics 1 = Elastic viscous plastic dynamics
kstrength	Integer	1	Determines pressure formulation 0 = Hibler (1979) parameterization 1 = Rothrock (1975) parameterization
evp_damping	Logical	.false.	If true, use damping procedure in evp dynamics (not supported).
snow_into_ocn	Logical	.false.	If true, snow on ridged ice falls into ocean (not supported).
advection	Character	'mpdata2'	Determines horizontal advection scheme. 'mpdata2' = second order advection 'upwind' = first order advection
grid_type	Character	'displaced_pole'	Determines grid type. 'displaced_pole' or 'rectangular' (not supported)

Table 8: Namelist Variables for File Names

Varible	Type	Default Value	Description
grid_file	Character	'data.domain.grid'	Input filename containing grid information.
kmt_file	Character	'data.domain.kmt'	Input filename containing land mask information.
pointer_file	Character	'ice.restart_file'	Pointer file that contains the name of the restart file.
incond_dir	Character	' '	Directory where netCDF initial condition file is output.
restart_dir	Character	' '	Directory where restart files are output.
history_dir	Character	' '	Directory where history files are output.
incond_file	Character	'incond'	Root name of netCDF output initial condition file.
dump_file	Character	'iced'	Prefix for output file containing restart information.
history_file	Character	'iceh'	Root name of history files.
mss_dir	Character	'null'	Directory on NCAR MSS where history and restart files will be written.
mss_rtpd	Integer	365	MSS retention period in days.
mss_pass	Character	' '	MSS password.
mss_rmlf	Integer	0	If true (=1), local files will be removed after being written to MSS.

kitd determines the scheme used to redistribute sea ice within the ice thickness distribution (ITD) as the ice grows and melts. The linear remapping scheme is the default and approximates the thickness distribution in each category as a linear function (Lipscomb (2001)). The delta function method represents $g(h)$ in each category as a delta function (Bitz et al. (2001)). This method can leave some categories mostly empty at any given time and cause jumps in the properties of $g(h)$.

kdyn determines the ice dynamics used in the model. The default is the elastic-viscous-plastic (EVP) dynamics Hunke and Dukowicz (1997). If kdyn is set to 0, the ice dynamics is inactive. In this case, ice velocities are not computed and ice is not transported. Since the initial ice velocities are read in from the restart file, the maximum and minimum velocities written to the log file will be non-zero in this case, but they are not used in any calculations.

The value of kstrength determines which formulation is used to calculate the strength of the pack ice. The Hibler (1979) calculation depends on mean ice thickness and open water fraction. The calculation of Rothrock (1975) is based on energetics and should not be used if the ice that participates in ridging is not well resolved.

evp_damping is used to control the damping of elastic waves in the ice dynamics. It is typically set to .true. for high-resolution simulations where the elastic waves are not sufficiently damped out in a small timestep without a significant amount of subcycling. This procedure works by reducing the effective ice strength that's used by the dynamics and is not a supported option.

The value of snow_into_ocean determines what happens to the snow on ice that is ridged. The default value is .false., so the snow cover remains on the ridged ice. If snow_into_ocean is .true., then the snow on ice that undergoes ridging is put into the ocean. This may cause a problem with the fresh water budgets; this is not a supported option.

advection determines the advection scheme used. The default is mpdata, which is second order accurate, but more computationally expensive. The upwind scheme is only first order accurate.

For coupled runs, both supported grids (gx3v4 and gx1v3) are 'displaced_pole'. The 'rectangular' option for a regular grid with constant latitude and longitude spacing is not supported.

Table 9: Namelist Variables for Prescribed Ice Model

Varible Name	Type	Default Value	Description
prescribed_ice	Logical	.false.	If true, run model with prescribed ice data.
prescribed_ice_file	Character	'prescribed_ice.nc'	Name of file with prescribed ice data.
prescribed_ice_climatology	Logical	.true.	If true, use AMIP climatology for prescribed ice model.

4.3.1.4 Namelist Variables for File Names The namelist parameters listed in Table 8 are for initial condition, restart, and history file and directory information. Information for NCAR MSS access is also in Table 8. During execution, the ice model reads grid and land mask information from the files grid_file and kmt_file that should be located in $EXEDIR. There are commands in *ice.setup.csh* that copy these files from the input data directory to $EXEDIR, rename them from *global_$ICE_GRID.grid* and *global_$ICE_GRID.kmt* to the default filenames shown in Table 8.

The namelist variable pointer_file is set to the name of the pointer file that contains the name of the restart file that will be read when model execution begins. This file resides in the $SCRIPTS directory and is created initially by the ice setup script but is overwritten every time a new restart file is created. It will contain the name of the latest restart file. The default filename *ice.restart_file* shown in Table 8 will not work unless some modifications are made to the ice setup script and a file is created with this name and contains the name of a valid restart file; this variable must be set in the namelist. More information on restart pointer files can be found in section 4.8.2.1.

incond_dir, restart_dir and history_dir are the directories where the initial condition file, the restart files and the history files will be written, respectively. These values are set at the top of the ice setup script and have been modified from the default values to meet the requirements of the CCSM2 filenaming convention. This allows each type of output file to be written to a separate directory. If the default values are used, all of the output files will be written to $EXEDIR.

incond_file, dump_file and history_file are the root filenames for the initial condition file, the restart files and the history files, respectively. These strings have been determined by the requirements of the CCSM2 filenaming convention, so the default values listed in Table 8 are not the same as those shown in Section 4.3.2.1. See sections 4.8.2 and 4.8.3 for an explanation of how the rest of the filename is created.

The namelist variables that contain the string 'mss' determine if ice model output files will be written to the NCAR MSS and where they will be stored. Currently, the ice model has the capability to write only initial condition, history and restart files to MSS. The *ice.log.$LID* files are also written to MSS, but not via the ice model. The data harvester (see CCSM2 User's Guide) will write these files to a */logs* directory. If the first four characters of mss_dir are set to 'null', no ice model output is written to MSS. If mss_dir is not set in the namelist, it will be set to the default value of 'null' in the ice model. As shown in example 1, mss_dir takes the value of $MSSDIR that is set in the main setup script. The MSS retention period mss_rtpd and password mss_pass are also set in the main script. If mss_rmlf is set to 1, each time a new history file is written to MSS, the previous history file is removed from *$EXEDIR/hist*.

4.3.1.5 Namelist Variables for Prescribed Ice Model The namelist variables for the prescribed ice model are shown in Table 9. As shown in the section 4.3.2.1, all of these variables are set via environment variables in the ice setup script. To run the prescribed ice model, $PRESCRIBED_ICE must be set to true at the top of *ice.setup.csh*. There are commands in *ice.setup.csh* that set $PICE_DATA to the name of the file containing the Atmospheric Model Intercomparison Project (AMIP) ice concentration data, and copy this file from the input data directory to $EXEDIR. The default filename *prescribed_ice.nc* shown in Table 9 will not work unless modifications are made to the ice setup script. If $PRESCRIBED_ICE_CLIM is set to true, the model will use a single annual cycle of monthly averaged climatological AMIP data repeatedly. If this parameter is set to false, the model will use a 19 year time series of monthly averaged AMIP observations. More details on how to set up the prescribed ice model and the input data can be found in sections 3.5 and 4.6.6.

Table 10: Namelist Variables for Ocean Mixed Layer Model

Varible Name	Type	Default Value	Description
`oceanmixed_ice`	Logical	.false.	If true, run model with ocean mixed layer model.
`oceanmixed_ice_file`	Character	'oceanmixed_ice.nc'	Name of file with ocean mixed layer data.
`oceanmixed_ice_sst_init`	Logical	.false.	If true, Jan 1 sst is read from forcing file.
`prntdiag_oceanmixed`	Logical	.false.	If true, print ocean mixed layer diagnostics.

4.3.1.6 Namelist Variables for Ocean Mixed Layer Model

The namelist variables for the ocean mixed layer model within the ice model are shown in Table 10. To use the slab ocean model, $OCEAN-MIXED_ICE must be set to true at the top of *ice.setup.csh*. There are commands in *ice.setup.csh* that will copy the grid dependent forcing file from the input data directory to $EXEDIR and rename it *oceanmixed_ice.nc*. If $OML_ICE_SST_INIT is set to true at the top of *ice.setup.csh*, the model will read the Jan 1 sea surface temperature from the forcing file. If this parameter is set to false, the model will read sea surface temperature and the freeze/melt potential from a restart file. More details on how to set up the slab ocean mixed layer model can be found in section 4.6.7.

4.3.2 Example Namelists

The following example is the default namelist taken directly from *ice.setup.csh*. Most of the variables in the namelist are determined from environment variables set in the main setup script (see Table 1) or prior to the namelist in the ice setup script (Table 2). Variables that are commonly changed directly in the namelist are the timestep `dt` and the number of subcycles per timestep in the ice dynamics `ndte`.

The second namelist controls what variables are written to the history file. The logical variables shown in this example are input variables from the atmosphere and ocean. Due to the reduced size of the buffer being sent to and received from the coupler, the values in these fields are not known everywhere, so they are not currently written to the history file. These fields can be retrieved from the ocean or atmosphere history files.

4.3.2.1 Example 1: Default Namelist Variables

```
cat << EOF >! ice_in
  &ice_nml
    runid       = '$CASE $CASESTR'
  , runtype     = '$RUNTYPE'
  , istep0      = 0
  , dt          = 3600.0
  , ndte        = 120
  , npt         = 99999
  , diagfreq    = 24
  , histfreq    = 'm'
  , dumpfreq    = 'y'
  , hist_avg    = .true.
  , restart     = $RESTART
  , print_points = .false.
  , kitd        = 1
  , kdyn        = 1
  , kstrength   = 1
  , evp_damping = .false.
```

```
  , snow_into_ocn = .false.
  , advection    = 'mpdata2'
  , grid_type    = 'displaced_pole'
  , grid_file    = 'data.domain.grid'
  , kmt_file     = 'data.domain.kmt'
  , incond_dir   = '$INIDIR/'
  , incond_file = '$CASE.csim.i.'
  , restart_dir  = '$RSTDIR/'
  , dump_file    = '$REST_ROOT'
  , history_dir  = '$HSTDIR/'
  , history_file = '$CASE.csim.h'
  , mss_dir      = '$MSSDIR/ice/ '
  , mss_rtpd     =  $MSSRPD
  , mss_pass     = '$MSSPWD'
  , mss_rmlf     = 0
  , prescribed_ice              = $PRESCRIBED_ICE
  , prescribed_ice_file         = '$PICE_DATA'
  , prescribed_ice_climatology = $PRESCRIBED_ICE_CLIM
  , pointer_file  = '$SCRIPTS/rpointer.$MODEL'
  , oceanmixed_ice              = $OCEANMIXED_ICE
  , oceanmixed_ice_file         = 'oceanmixed_ice.nc'
  , oceanmixed_ice_sst_init     = $OML_ICE_SST_INIT
  , prntdiag_oceanmixed         = .false.
/

  &icefields_nml
    f_Fswdn   = .false.
  , f_Flwdn   = .false.
  , f_snow    = .false.
  , f_rain    = .false.
  , f_sst     = .false.
  , f_sss     = .false.
  , f_uocn    = .false.
  , f_vocn    = .false.
  , f_frzmlt  = .false.
  , f_strltx = .false.
  , f_strlty = .false.
  , f_Tref    = .false.
  /
```

4.4   Model Input Datasets

The D configuration requires several datasets, many of which are grid-dependent. The following example is for the supported T42_gx1v3 grid. When building the D configuration with datm5, the following files will be copied to the $EXEROOT directory from the input data directory:

```
$EXEROOT/atm/f20.006.cam2.h1.0006-01-01-00000.nc
$EXEROOT/cpl/map_a2of.nc
$EXEROOT/cpl/map_a2os.nc
$EXEROOT/cpl/map_o2af.nc
$EXEROOT/cpl/map_r2o.nc
$EXEROOT/ice/data.domain.grid
$EXEROOT/ice/data.domain.kmt
```

```
$EXEROOT/ice/rest/iced.0001-01-01.gx1v3
$EXEROOT/lnd/f20.006.cpl5.ha.0006-xx.nc
$EXEROOT/ocn/domain.gx1v3_010723.nc
$EXEROOT/ocn/sst_clim_hurrell_1x1_020827.nc
```

The files in the coupler (cpl) directory map information between the T42 and the gx1v3 grids. Some of the files listed above have been renamed in the process of being moved to the $EXEROOT directory. To find the name of the original file, look in the setup script of the appropriate component.

To run the M configuration with latm5 and the ocean mixed layer within the ice model on the T62_gx1v3 grid using one year of forcing data from 1985, the following files will be copied to the $EXEROOT directory:

```
$EXEROOT/atm/dn10.1985.nc
$EXEROOT/atm/lwdn.1985.nc
$EXEROOT/atm/prec.1985.nc
$EXEROOT/atm/q_10.1985.nc
$EXEROOT/atm/swdn.1985.nc
$EXEROOT/atm/swup.1985.nc
$EXEROOT/atm/t_10.1985.nc
$EXEROOT/atm/u_10.1985.nc
$EXEROOT/atm/v_10.1985.nc
$EXEROOT/cpl/map_a2of.nc
$EXEROOT/cpl/map_a2os.nc
$EXEROOT/cpl/map_o2af.nc
$EXEROOT/cpl/map_r2o.nc
$EXEROOT/ice/data.domain.grid
$EXEROOT/ice/data.domain.kmt
$EXEROOT/ice/oceanmixed_ice.nc
$EXEROOT/ice/rest/iced.0001-01-01.gx1v3
$EXEROOT/lnd/lsmh_0016.nc
$EXEROOT/ocn/domain.gx1v3_010723.nc
$EXEROOT/ocn/sst_clim_hurrell_1x1_020827.nc
```

Note that this is only one particular example of using latm5. See the data model User's Guide for more information. The input files required for CSIM4 are as follows:

- `global_${ICE_GRID}.grid` is a binary file containing grid information and is renamed *data.domain.grid* when it is copied to the $EXEROOT directory.

- `global_${ICE_GRID}.kmt` is a binary file containing land mask information and is renamed *data.domain.kmt*.

- `iced.0001-01-01.${ICE_GRID}` is a binary file containing initial condition information.

Depending on `${ICE_GRID}` selected in the main driver script, the appropriate `global*` files will be copied and renamed in *$EXEROOT/ice*. The restart file will be copied to *$EXEDIR/rest* which is set in the ice setup script. Currently, only gx3v4 and gx1v3 grids are supported for the ice and ocean models.

To run the prescribed ice model, an additional data file is necessary and depends on the chosen grid. For the high resolution grid:

- `sst_clim_hurrell_gx1v3_020827.nc` contains ice concentration and SST data.

is required and for the low resolution grid

- `sst_clim_hurrell_gx3v4_020827.nc`

is required.

These datasets contain one year of monthly averaged Hurrell-modified Hadley SST climatology, 1949-2001. `PRESCRIBED_ICE = .true.` and `PRESCRIBED_ICE_CLIM = .true.` must be set at the top of *ice.setup.csh* for this data set to be used.

Another data file is necessary to use the ocean mixed layer within the ice model, depending on the specified grid:

- `pop_frc_gx1v3_010815.nc` contains monthly averaged ocean forcing from POP output.

- `pop_frc_gx3v4_020801.nc` same as above, but for the gx3v4 grid.

This file is renamed *oceanmixed_ice.nc* when it is copied into *$EXEROOT/ice*.

4.5   The Four Run Types

The environment variable $RUNTYPE set in the main setup script specifies the state in which the model will begin a run. This variable has four options, 'startup', 'hybrid', 'continue' or 'branch', with 'startup' being the default setting. The four options are described in this section.

4.5.1   Startup Runs

If $RUNTYPE is set to 'startup', the model will start from some arbitrary baseline state that need not be associated with any previous run. Startup runs are typically initialized using a $BASEDATE of 0001-01-01, although there is no dependence of these runs on previous case names or calendar dates. The ice model uses the $BASEDATE sent from the coupler on a startup run. The ice setup script will create a pointer file named *rpointer.ice* with the name of the initial restart file in it using $BASEDATE. The date read in from the restart file will be written over by $BASEDATE. The primary environment variables for a startup run are:

```
CASE       test.a1        #new case name
RUNTYPE    startup
BASEDATE   0001-01-01     #start date for initial run
```

Startup runs can start from a $BASEDATE other then 0001-01-01. Startup runs can also be initialized using data created within the ice model, as described in the next section.

4.5.1.1   Using Initial Conditions from within CSIM4   Initial conditions can be calculated within the ice model in a subroutine called init_state in *ice_init.F*. Here, the ocean surface is initialized at the freezing point everywhere north of 40 degrees and south of -40 degrees latitude, allowing ice to form everywhere in these regions. While running the ice model with a data ocean will melt any extra ice during the first year of integration, is not recommended that these initial conditions be used when running the ice model coupled to an active ocean model. The advantage of using this input is that it is not grid or land mask dependent. To use these initial conditions, set RESTART in *ice.setup.csh*:

```
set RESTART = .false.
```

4.5.1.2   Using Initial Conditions from Input Data Files   The input data file for the ice model is a binary restart file from a previous run. If $RUNTYPE is set to 'startup', the ice setup script is will read the initial condition file from the input data directory, *iced.0001-01-01.$ICE_GRID* regardless of $BASEDATE. For 'continue' runs, the input data file is determined by the filename set in the restart pointer file (see section 4.8.2.1). If $RUNTYPE is 'branch' or 'hybrid', the ice setup script will attempt to file a restart file with $REFDATE in the filename.

For the startup and hybrid setups, it is not necessary that the date inside the binary file match the date in the filename, since the $BASEDATE received from the coupler will override the date in the restart file. This will not be true for continue and branch runs, where the $BASEDATE is set in the model using the date in the restart file. Initializing the model using a restart file from an equilibrium run will result in a more physically reasonable scenario than the initial conditions set within CSIM4. The conditions in the initial

restart file provided with CSIM4 are from an active ice only equilibrium run using modified NCEP forcing. The drawbacks are that the data is binary, difficult to edit, and is date and grid dependent. A restart file will be used as initial conditions if

```
set RESTART = .true.
```

in *ice.setup.csh*.

4.5.2 Hybrid Runs

A hybrid run should be used when a collection of arbitrary initial/restart files from previous runs is desired to start the model. A new $CASE and $BASEDATE can be specified. The ice model receives $BASEDATE from the coupler. The date read in from the restart file will be written over by $BASEDATE. The climate simulated by the hybrid run should be continuous compared to the previous run, but an exact restart will not be produced. If all components start from the same reference case and date, $REFCASE and $REFDATE can be used to specify the previous date and case. Generally, a hybrid run will require some modifications to the scripts by the user, especially if some components are starting from inconsistent runs or dates. The primary environment variables for a startup run are:

```
CASE       test.a1       #new case name
RUNTYPE    hybrid
BASEDATE   0001-01-01    #date where to start run
REFCASE                  #reference case for initial/restart data
REFDATE    0001-01-01    #date in REFCASE to obtain initial/restart data
```

4.5.3 Continue Runs

A continue run is an exact continuation of a previous run. This means that the run will produce a bit-for-bit answer as if the existing run had not stopped. In this case, it is assumed that $CASE and $BASEDATE have not changed and that the date is continuous from the previous run. For a continue run, the only change required in the main script is to set:

```
RUNTYPE    continue
```

A change in the $BASEDATE is not necessary for a continue run, since the date is read in from the restart files, and the coupler checks to see that the dates between the component models match.

4.5.4 Branch Runs

Branch runs will continue a previous run but with a new case name and possibly a new start date, where the previous run has created a consistent set of restart files. Branch runs differ from continuation runs in that they are typically used to perform sensitivity studies, or to modify output streams, branching off a previous run. The new case will restart exactly, as in a continuation run, if no source code or input namelist parameters have changed. For a branch run, $BASEDATE must remain the same as that in the previous run, and the following environment variables must be set in the the main script:

```
CASE       new case name
RUNTYPE    branch
BASEDATE   date to start this run
REFCASE    reference case to branch from
REFDATE    date in REFCASE to obtain restart data
```

4.6   Modifying the Input Variables

This section provides instructions for changing the appropriate environment variables in the main and ice setup scripts to set up some of the common configurations.

4.6.1 Changing Grid Resolution

The grid is set using the environment variable $GRID in the main setup script. The available grids are gx3v4 and gx1v3 for the ocean and ice models, which must be run on the same grid. The options are T31, T42 and T62 for the atmosphere and land models, which also must be on the same grid. The default grid is T42_gx1v3, which is the T42 grid for the atmosphere/land models and the gx1v3 grid for the ice/ocean components. For a faster simulation, this can be changed to T31_gx3v4, which is the coarse resolution grid for the ice/ocn. T62 is not a supported grid for the active atmosphere and land models, but is a supported resolution for latm5. See section 3.4 for details on how to use this grid.

4.6.2 Changing the Number of Processors

The number of tasks, or processors, used by each component is set in the main script using the variable $NTASKS. All data models are run using a single task. For the T42_gx1v3 grid, the ice model is typically run on 8 or 16 tasks. Note that the number of tasks used by the ice model must divide evenly into the grid. For example, running the ice model with 8 tasks on the gx3v4 grid wil result in an error, since 8 does not divide evenly into the 100 longitude points. When this happens, the model will stop with an error message written to the log file.

To modify the number of tasks, the keyword `task_geometry` and the environment variable $NTASKS need to be changed in the main script. The `task_geometry` keyword allows groups of tasks to run together on the same node. Currently, on the IBM at NCAR, a node is a group of four processors. The following examples are for the D configuration with the components in the following order as specified in the main script:

```
 set SETUPS = ( datm  dlnd  ice  docn  cpl )
```

For four processors, the task geometry and NTASKS are

```
# @ task_geometry = {(0,1,6)(2,3,4,5)(7)}
set NTASKS = (     1    1    4    1    1 )  # use NTASK = 1 for data model
set NTHRDS = (     1    1    1    1    4 )  # use NTHRD = 1 for data model
```

Each number in the above `task_geometry` represents a task ID number, with each set of parenthesis defining a node. In this example, a job with eight tasks will run on three different nodes. datm, dlnd, docn will share a node, the ice model will use four tasks assigned to a single node and the coupler will use a single node and spawn four threads. Four tasks for the ice model is sufficient for running on the gx3v4 grid. To run the ice model using eight processors, set:

```
# @ task_geometry = {(0,1,10)(2,3,4,5)(6,7,8,9)(11)}
set NTASKS = (     1    1    8    1    1 )  # use NTASK = 1 for data model
```

In this example, the ice model will use eight tasks assigned to two nodes. For sixteen processors set

```
# @ task_geometry = {(0,1,18)(2,3,4,5)(6,7,8,9)(10,11,12,13)(14,15,16,17)(19)}
set NTASKS = (     1    1    16    1    1 )  # use NTASK = 1 for data model
```

The prescribed ice model has a complex task geometry due to the active atmosphere and land models that are threaded.

```
# @ task_geometry = {(0)(1)(2)(3)(4)(5)(6)(7)(8)(9)(10)(11,12,13,14)(15,16,17,18)
                     (19,20,21,22)(23,24,25,26)(27)(28)}
set SETUPS = (   atm  lnd  ice docn  cpl )
set NTASKS = (    8    3    16    1    1 )  # use NTASK = 1 for data model
set NTHRDS = (    4    4    1    1    4 )  # use NTHRD = 1 for data model
```

In this case, the atmosphere runs threaded on tasks 0 through 7, and the land runs threaded on tasks 8 through 10. The ice model uses tasks 11 through 26 on spread over four nodes. docn and cpl are set up as in the previous examples.

4.6.3 Changing the Ice Model Timestep

The ice model timestep is controlled by the variable `dt` in the `&ice_mnl` namelist in *ice.setup.csh*. The units are in seconds. See section 4.3.1 for a discussion on how to change the timestep and the dynamics subcycling.

4.6.4 Changing the Length of the Model Run

The length of the model run is controlled by the coupler. To change the length of the model run, change `stop_n` in the `$MODEL.stdin` namelist in *cpl.setup.csh*. This parameter will be in the units set by `stop_option`, which is typically 'ndays'. See the section 4.1.3.6 and the coupler documentation for the other options and examples.

4.6.5 Changing the Number of Ice Thickness Categories

The number of ice thickness categories affects files in three places:

- $NCAT in the ice setup script

- The input template files *ice_model_size.F.$RESx$NCAT* in *models/ice/csim4/src/input_templates*

- The initial condition (restart) file in the input file directory

The number of ice thickness categories is set in *ice.setup.csh* using the variable called $NCAT. The default value is 5 categories. The script variable $NCAT is used to determine which module *ice_model_size.F.$RESx$NCAT* is copied to $OBJDIR before compilation where it is renamed *ice_model_size.F*. $RES is the resolution of the grid, 100x116 (gx3v4) and 320x384 (gx1v3) for low and high resolution grids, respectively. The input templates are located in *models/ice/csim4/src/input_templates* for 1, 3, 5 and 10 thickness categories on the default grid sizes of 100x116 and 320x384.

To create an input template with a number of categories other than the above values, copy an existing template that has the desired grid. Change `ncat` to the appropriate number of categories. `ntilay`, the total number of mid-layer temperature points in all ice thickness categories, must also be changed. The number of these points in each category is set in *ice_vthermo.F* in the subroutine init_thermo and depends on the number of categories chosen. The number of layers assigned to each category in the array `nilay` will have to be added to get the total number of layers. For example, for seven thickness categories, ntilay will be 22: two layers in each of the first three categories, and four layers in each of the last four categories.

The information in the initial condition file is dependent on the number of ice thickness categories. An initial condition file exists only for the default case of 5 ice thickness categories. To create an initial condition file for a different number of categories, the following steps should be followed:

- Set $NCAT to the desired number of categories in *ice.setup.csh*.

- Set the namelist variable `dumpfreq='m'` to print out restart files monthly.

- Create an input template (or use an existing one) with the desired number of categories.

- Run the M configuration (see section 3.4) to equilibrium using the initial conditions within the ice model (see section 4.5.1.1).

- The last restart file can be used as an initial condition file.

- Change the name of the last restart file to iced.0001-01-01.$GRID.

- Copy the file into the input data directory or directly into $RSTDIR.

Note that the date printed inside the binary restart file will not be the same as 0001-01-01. If the $RUNTYPE is 'startup', $BASEDATE will be the starting date and the date inside the file will not be used.

4.6.6 Using the Prescribed Ice Model

The prescribed ice model is typically used to do coupled runs with an active atmosphere. It reads in monthly averaged ice concentrations that are used as a fixed lower boundary condition. This option will run on either the gx3v4 or the gx1v3 grid and cannot be run with the ocean mixed layer model within the ice model.

Two datasets are available to force the prescribed ice model. Both are in netCDF format, based on SSMI data, and contain sea surface temperature. The first is AMIP data that is available in a one year climatology and a 243 month time series. The second dataset was created by Jim Hurrell at NCAR and is currently available in a one year climatology. To use the prescribed ice model with either dataset, verify that the following variables are set in the main script:

- set GRID to T42_gx1v3 or T31_gx3v4

- `setenv BASEDATE 0001-01-01`

and that the following change is made in the ice setup script:

- `set $PRESCRIBED_ICE = .true.`

The following sections describe what changes need to be made to the ice setup script to use the desired dataset.

4.6.6.1 Using the Hurrell dataset One advantage of using the Hurrell dataset is that it was used to create the CAM2 forcing datasets used in datm5, so they are consistent. By default, the prescribed ice model will use the Hurrell dataset and will cycle through the single climatological year repeatedly. The following variable should be set in *ice.setup.csh*:

- `set $PRESCRIBED_ICE_CLIM = .true.`

Since there is currently no time series for the Hurrell data, $PRESCRIBED_ICE_CLIM must be set to `.true.` in this case. The ice setup script will exit if `PRESCRIBED_ICE_CLIM = .false.` and the Hurrell dataset is selected. The file names for this data set are already set using the variable $PICE_DATA:

```
if ($ICE_GRID == 'gx1v3') \
    set PICE_DATA = sst_clim_hurrell_gx1v3_020827.nc
if ($ICE_GRID == 'gx3v4') \
    set PICE_DATA = sst_clim_hurrell_gx3v4_020827.nc
```

4.6.6.2 Using the AMIP dataset The climatological and time series AMIP data are contained in a single file. The first two years of the dataset are climatology, represented by years 1976 and 1977. Year 1978 is a transition period where the data changes gradually so that by December of 1978, it will have reached the monthly observed data in a smooth fashion. The time series of observed data begins in January of 1979.

To use the AMIP forcing the following changes will have to be made to $PICE_DATA in the ice setup script:

```
if ($ICE_GRID == 'gx1v3') \
    set PICE_DATA = AMIP_bcgx1v3_1976-1996_010817.nc
if ($ICE_GRID == 'gx3v4') \
    set PICE_DATA = AMIP_bcgx3v4_1976-1996_020801.nc
```

There two ways to force the prescribed ice model with AMIP data. The first option is to use use a single annual cycle of the climatological repeatedly. The second is to use the 19 year time series of monthly averaged observations.

To use the climatological data:

- `set $PRESCRIBED_ICE_CLIM = .true.` in *ice.setup.csh*

To use the 19 year monthly averaged observed time series:

- `setenv BASEDATE 1977-01-01` in main driver script

- `set $PRESCRIBED_ICE_CLIM = .false.` in *ice.setup.csh*

The ice setup script will exit if `$PRESCRIBED_ICE_CLIM = .false.` and `$BASEDATE` is not set correctly.

4.6.7 Using the Ocean Mixed Layer Model within CSIM4

An ocean mixed layer model has been incorporated into the ice model since docn5 does not allow frazil ice growth to occur. This is due to the minimum ocean mixed layer temperature being fixed at the freezing point. It is a simple slab ocean mixed layer model that is forced using output from a POP ocean simulation. More details on the physics of the ocean mixed layer model can be found in the Physics Documentation. This option can be run with the gx3v4 or the gx1v3 grid. To use the mixed layer model:

- `set $OCEANMIXED_ICE = .true.` in *ice.setup.csh*

The variable $OML_ICE_SST_INIT determines the initial sea surface temperature. If the run is a startup run, this variable is set to true in the ice setup script, and the January 1 value of the sea surface temperature is read from the POP forcing file. Thereafter, the value of $OML_ICE_SST_INIT is set to false, and sea surface temperature and the freeze/melt potential is read from a restart file. The ocean mixed layer model cannot be run concurrently with the prescribed ice model. When the slab ocean mixed layer within the ice model is used, the data that is received from the coupler from the ocean component (docn or POP) is overwritten by the values calculated by the ocean mixed layer. Therefore, it is not appropriate to use the ocean mixed layer option with an active ocean model.

4.7 Information Sent to and Received from Coupler

The ice model receives ocean and atmosphere states and fluxes from the coupler and returns aggregate ice states and fluxes. Tables 2 and 3 in the Scientific Documentation list the fields that are exchanged. The flux coupler multiplies the fluxes it receives from the ice model by the ice area, so before fluxes are sent to the coupler, they are divided by the ice area (see module *ice_scaling.F*). To increase the performance of CCSM2, a packed buffer is exchanged with the coupler that contains a reduced set of information.

How this data is reduced depends on the model configuration. For the fully coupled (B) configuration, information is sent to the coupler only in ocean grid cells that have a non-zero ice area (see module *ice_coupling.F*). The prescribed ice model, or F configuration, sends data in ocean grid cells where the monthly averaged ice concentration bracketing the current date and the next monthly average are non-zero. For the M configuration, data is sent in ocean grid points for latitudes south of -40 degrees and north of 35 degrees (see module *ice_grid.F*). The land mask is modified to screen out these points; the area between these latitudes appears as land in the history files.

The coupler returns information to the ice model via a packed vector. Beginning at the equator, the coupler searches for the first grid cell where either the ice concentration is greater then zero, there is potential to form ice, or data was sent by the ice model. An additional five grid points are then added. This is done in both hemispheres.

The packing of the buffer exchanged with the coupler and the multiplication of the fields received from the ice model by the ice area in the coupler affect the information written to the history files. These effects are discussed in the next section.

4.7.1 Effects of Packed Buffer on History File Data

Before fields are received from the coupler, the corresponding arrays in the ice model are set to zero and only the active cells are filled in. As a result, information sent to the ice model is not known everywhere, but all grid values are written to the history file. See section 4.8.3.1 for information on how fields are averaged before being written to the history file. The affected fields are

- Fswdn, Flwdn - downwelling shortwave and longwave

28

- snow, rain - snowfall and rainfall

- sst, sss - sea surface temperature and salinity

- uocn, vocn - ocean velocities

- strtltx, strtlty - zonal and meridional sea surface tilt

- Tref - reference temperature

These fields can be retrieved from the ocean or atmosphere history files, so they have been removed from the ice history files via the namelist. If performance is not a problem, the buffer can be sent to the coupler unpacked, then all of the above fields are valid.

The coupler multiplies fluxes it receives from the ice model by the ice area. Before sending fluxes to the coupler, they are divided by the ice area in the ice model. These are the fluxes that are written to the history files, they are not what affects the ice, ocean or atmosphere, nor are they useful for calculating budgets. The division by the ice area also creates large values of the fluxes at the ice edge. The affected fields are:

- Flat, Fsens - latent and sensible heat fluxes

- Flwout - outgoing longwave

- evap - evaporative water flux

- Fresh - ice/ocn fresh water flux

- Fhnet - ice/ocn net heat flux

- Fswabs - snow/ice/ocn absorbed solar flux

When applicable, two of the above fields will be written to the history file: the value of the field that is sent to the coupler (divided by ice area) and a value of the flux that has been multiplied by ice area (what affects the ice). Fluxes multiplied by ice area will have the suffix _aice appended to the variable names in the history files. Fluxes sent to the coupler will have "sent to coupler" appended to the long_name. Fields of rainfall and snowfall multiplied by ice area are written to the history file, since the values are valid everywhere and represent the precipitation rate on the ice cover.

4.8 Output Data

The ice model produces three types of output data. A file containing ASCII text, also known as a log file, is created for each run that contains information about how the run was set up and how it progressed. A series of binary restart files necessary to continue the run are created. A series of netCDF history files containing gridded instantaneous or time-averaged output are generated during a run. These are described below.

4.8.1 Stdout Output

Diagnostics from the ice model are written to an ASCII file that consists of a copy of the ice setup script file, information from the compilation, a record of the input parameters, and how hemispherically averaged, maximum and minimum values are evolving with the integration. Certain error conditions detected within the ice setup script or the ice model will also appear in this file. Upon the completion of the simulation, some timing information will appear at the bottom of the file. The file name is of the form `ice.log.$LID`, where `$LID` is a timestamp for the file ID. It resides in the `$EXEROOT` directory. The frequency of the diagnostics is determined by the namelist parameter `diagfreq`.

4.8.2 Restart Files

Restart files contain all of the initial condition information necessary to restart a previous simulation. These files are in a standard IEEE 64 bit binary format. A restart file is not necessary for an initial run, but is highly recommended. The initial conditions that are internal to the ice model produce an unrealistic ice cover that an uncoupled ice model will correct in several years. The initial conditions from a restart file are created from an equilibrium solution, and provide more realistic information that is necessary if coupling to an ocean model. The frequency at which restart files are created is controlled by the namelist parameter `dumpfreq`. These restart files contain the ice state variables and the land mask information. The names of these files are proceeded by the namelist parameter `dump_file` and, by default are written out yearly to the directory `$EXEROOT/rest`. The names of the restart files follow the CCSM2 Output Filename Requirements. The form of the restart file names are as follows:

$CASE.csim.r.yyyy-mm-dd-sssss

For example, the file *$CASE.csim.r.0002-01-01-00000* would be written out at the end of year 1, month 12. A file containing the name of a restart file is called a restart pointer file. This filename information allows the model simulation to continue from the correct point in time, and hence the correct restart file. These pointer files are described in the next section.

4.8.2.1 Restart Pointer Files

A pointer file is an ascii file named *rpointer.ice* that contains the path and filename of the latest restart file. The model uses this information to find a restart file from which initialization data is read. The pointer files are written to and then read from the */scripts/$CASE* directory. For 'startup' runs, a pointer is created by the ice setup script using $BASEDATE to create the file name. Pointer files are also created by the ice model during execution at the frequency determined by namelist variables set in the ice model `dumpfreq` and the coupler `rest_freq`. Whenever a restart file is written, the existing restart pointer file is overwritten. The namelist variable `pointer_file` contains the name of the pointer file. Pointer files seldom need editing. The contents are usually maintained by the setup script and the the component model.

4.8.3 History Files

History files contain gridded data values written at specified times during a model run. The netCDF file names are prepended by the character string given by `history_file` in the `ice_nml` namelist. This character string has been set according to CCSM2 Output Filename Requirements. If `history_file` is not set in the namelist, the default character string 'iceh' is used. The user can specify the frequency at which the data are written. Options are also available to record averaged or instantaneous data. The form of the history file names are as follows:

Yearly averaged: *$CASE.csim.h.yyyy.nc*
Monthly averaged: *$CASE.csim.h.yyyy-mm.nc*
Weekly averaged: *$CASE.csim.hw.yyyy-mm-dd-sssss.nc*
Daily averaged: *$CASE.csim.h.yyyy-mm-dd.nc*
Instantaneous (`histfreq` = 'y', 'm', 'w' or 'd'): *$CASE.csim.h.yyyy-mm-dd-sssss.nc*
Instantaneous (written every dt, `histfreq` = 1): *$CASE.csim.h.yyyy-mm-dd-sssss.nc*

$CASE is set in the main setup script. All history files are written to the directory *$EXEDIR/hist*. Changes to the frequency and averaging will affect all output fields. The best description of the history data comes from the file itself using the netCDF command `ncdump -h filename.nc`. Variables containing grid information are written to every file and are listed in Table 11. In addition to the history files, a netCDF file containing a snapshot of the initial ice state is created at the start of each run. The file name is *$CASE.csim.i.yyyy-mm-dd-sssss.nc* and is written to *$EXEDIR/init*.

4.8.3.1 Information on Averaged Fields In computing the monthly averages for output to the history files, most arrays are zeroed out before being filled with data. These zeros are included in the monthly averages where there is no ice. For some fileds, this is not a problem, for example, ice thickness and ice area. For other fields, this will result in values that are not representative of the field when ice is present. Some of the fields affected are:

- Flat, Fsens - latent and sensible heat fluxes

- albsni - snow/ice broadband albedo

- evap - evaporative water flux

- Fhnet - ice/ocn net heat flux

- Fswabs - snow/ice/ocn absorbed solar flux

- strairx, strairy - zonal and meridional atm/ice stress

- strcorx, strcory - zonal and meridional coriolis stress

In some cases, a non-zero value is set where there is no ice. For example, Tsfc has the freezing point averaged in, and Flwout has σT_f^4 averaged in. At lower latitudes, these values can be erroneous.

To aid in the interpretation of the fields, a field called *ice_present* is written to the history file. It contains information on the fraction of the time-averaging interval when any ice was present in the grid cell during the time-averaging interval in the history file. This will give an idea of how many zeros were included in the average.

4.8.3.2 Changing Frequency and Averaging The frequency at which data are written to a history file as well as the interval over which the time average is to be performed is controlled by the namelist variable `histfreq`. Data averaging is invoked by the namelist variable `hist_avg`. The averages are constructed by accumulating the running sums of all variables in memory at each timestep. The options for both of these variables are described in Table 5. If `hist_avg` is true, and `histfreq` is set to monthly, for example, monthly averaged data is written out on the last day of the month.

4.8.3.3 Changing Content To remove a field from this list, add the name of the logical variable associated with that field to the `&icefields_nml` namelist in the *ice.setup.csh* file and assign it a value of `.false.`. For example, to remove ice thickness and snow cover from the history file, add

```
&icefields_nml
    f_hi  =  .false.
  , f_hs  =  .false.
/
```

to the namelist. When all fields listed in Tables 5 and 11 are written to the history file, the file size is 39.3 MB. When the twelve fields received from the coupler are removed (see section 4.7.1), the file size is 33 MB.

4.8.3.4 Changing Directory Location By default, the history files will be written to the directory *$EXEDIR/hist*. Modifications to $EXEDIR can be made in the main setup script. The directory *hist* is appended to $EXEDIR in the ice setup script so that the ice model history files are written into a separate directory.

There is an option to write the restart and history files to NCAR's Mass Storage System (MSS). In the main setup script, set

```
setenv MSSDIR   mss:/$MSSNAME/csm/$CASE        # MSS directory path name
setenv MSSRPD   365                            # MSS file retention period
```

$MSSNAME is set to your login name in capital letters. This will write history and restart files to the MSS with a one year retention period. The password is set to your login name:

Table 11: Time and Grid Information Written to History File

Field	Description	Units
time	model time	days
time_bounds	boundaries for time-averaging interval	days
TLON	T grid center longitude	degrees
TLAT	T grid center latitude	degrees
ULON	U grid center longitude	degrees
ULAT	U grid center latitude	degrees
tmask	ocean grid mask (0=land, 1=ocean)	
tarea	T grid cell area	m^2
uarea	U grid cell area	m^2
dxt	T cell width through middle	m
dyt	T cell height through middle	m
dxu	U cell width through middle	m
dyu	U cell height through middle	m
HTN	T cell width North side	m
HTE	T cell width East side	m
ANGLET	angle grid makes with latitude line on T grid	radians
ANGLE	angle grid makes with latitude line on U grid	radians
ice_present	fraction of time-averaging interval that any ice is present	

```
setenv MSSPWD    $LOGNAME                        # MSS file write password
```

To avoid writing files to the MSS, set:

```
setenv MSSDIR    null:/dev/nul                   # MSS directory path name
setenv MSSRPD    0                               # MSS file retention period
```

Table 12: Fields Written to History File

Logical Variable	Description	Units
f_hi	grid box mean ice thickness	m
f_hs	grid box mean snow thickness	m
f_Tsfc	snow/ice surface temperature	C
f_aice	ice concentration (aggregate)	%
f_aice1	ice concentration (category 1)	%
f_aice2	ice concentration (category 2)	%
f_aice3	ice concentration (category 3)	%
f_aice4	ice concentration (category 4)	%
f_aice5	ice concentration (category 5)	%
f_u	zonal ice velocity	cm s^{-1}
f_v	meridional ice velocity	cm s^{-1}
f_Fswdn	downwelling solar flux	W m^{-2}
f_Flwdn	downwelling longwave flux	W m^{-2}
f_snow	snow fall rate received from coupler	cm day^{-1}
f_snow_aice	snow fall rate on ice cover	cm day^{-1}
f_rain	rain fall rate received from coupler	cm day^{-1}
f_rain_aice	rain fall rate on ice cover	cm day^{-1}
continued on next page		

continued from previous page		
f_sst	sea surface temperature	C
f_sss	sea surface salinity	g kg^{-1}
f_uocn	zonal ocean current	cm s^{-1}
f_vocn	meridional ocean current	cm s^{-1}
f_frzmlt	freeze/melt potential	W m^{-2}
f_Fswabs	absorbed solar flux sent to coupler	W m^{-2}
f_Fswabs_aice	absorbed solar flux in snow/ocn/ice	W m^{-2}
f_albsni	snow/ice broad band albedo	%
f_Flat	latent heat flux sent to coupler	W m^{-2}
f_Flat_aice	ice/atm latent heat flux	W m^{-2}
f_Fsens	sensible heat flux sent to coupler	W m^{-2}
f_Fsens_aice	ice/atm sensible heat flux	W m^{-2}
f_Flwout	outgoing longwave flux sent to coupler	W m^{-2}
f_Flwout_aice	ice/atm outgoing longwave flux	W m^{-2}
f_evap	evaporative water flux sent to coupler	cm day^{-1}
f_evap_aice	ice/atm evaporative water flux	cm day^{-1}
f_Tref	reference temperature	C
f_growb	basal ice growth	cm day^{-1}
f_frazil	frazil ice growth	cm day^{-1}
f_snoice	snow-ice formation	cm day^{-1}
f_meltb	basal ice melt	cm day^{-1}
f_meltt	surface ice melt	cm day^{-1}
f_meltl	lateral ice melt	cm day^{-1}
f_Fresh	ice/ocn fresh water flux sent to coupler	cm day^{-1}
f_Fresh_aice	ice/ocn fresh water flux	cm day^{-1}
f_Fhnet	ice/ocn net heat flux sent to coupler	W m^{-2}
f_Fhnet_aice	ice/ocn net heat flux	W m^{-2}
f_strairx	zonal atm/ice stress	N m^{-2}
f_strairy	meridional atm/ice stress	N m^{-2}
f_strtltx	zonal sea surface tilt	m m^{-1}
f_strtlty	meridional sea surface tilt	m m^{-1}
f_strcorx	zonal coriolis stress	N m^{-2}
f_strcory	meridional coriolis stress	N m^{-2}
f_strocnx	zonal ocean/ice stress	N m^{-2}
f_strocny	meridional ocean/ice stress	N m^{-2}
f_strintx	zonal internal ice stress	N m^{-2}
f_strinty	meridional internal ice stress	N m^{-2}
f_strength	compressive ice strength	N m^{-1}
f_divu	velocity divergence	% day^{-1}
f_shear	strain rate	% day^{-1}
f_opening	lead opening rate	% day^{-1}
f_sig1	normalized principal stress component	
f_sig2	normalized principal stress component	
f_daidtt	area tendency due to thermodynamics	% day^{-1}
f_daidtd	area tendency due to dynamics	% day^{-1}
f_dvidtt	ice volume tendency due to thermo.	cm day^{-1}
f_dvidtd	ice volume tendency due to dynamics	cm day^{-1}

5   Trouble-shooting

The coupler does not compile on an SGI Origin 2000 using MIPSpro_7.2.1. Try using a more recent version of the compiler, such as MIPSpro_7.3.1.1.

5.1 Enabling the Debugger

This section explains how to set some compiler options for debugging the ice component on the NCAR IBM. Modifications need to be made to two files. In *ice.setup.csh*, the line `setenv DEBUG TRUE` needs to be added. In the first gmake command line, add `DEBUG=$DEBUG` after `THREAD=$THREAD`. Comment out the second gmake command line so that the `*.f` files are not deleted.

In the */models/bld/Macros.AIX* file there is a line with compiler options for the ice model under csim. Add the options `-g` and `-qsigtrap=xl__trcedump` to the `FFLAGS` macro under `csim`. Before running the model, be sure to delete the files in the `$OBJROOT/ice/obj` directory so that the ice code will be recompiled.

If a core file is created, it will be in the `$EXEROOT` directory, and the executable will be in `$EXEROOT`/ice. To look at the core file, type 'dbx `$EXEROOT`/ice/ice core' in the directory where the executable file is located, replacing `$EXEROOT` with the appropriate path name. Useful information may also appear in the `poe.stderr*` file in the directory where the main setup script is located.

References

Bitz, C. M., M. Holland, A. J. Weaver and M. Eby, 2001: Simulating the ice-thickness distribution in a coupled climate model. *J. Geophys. Res.*, **106**, 2441–2464.

Hibler, W. D., 1979: A dynamic thermodynamic sea ice model. *J. Phys. Oceanogr.*, **9**, 815–846.

Hunke, E. C., 2001: Viscous-plastic sea ice dynamics with the evp model: Linearization issues. *J. Comp. Phys.*, **170**, 18–38.

Hunke, E. C. and J. K. Dukowicz, 1997: An elastic-viscous-plastic model for sea ice dynamics. *J. Phys. Oceanogr.*, **27**, 1849–1867.

Hunke, E. C. and W. H. Lipscomb, 2002: *CICE: the Los Alamos sea ice model, documentation and software User's Manual*. T-3 Fluid Dynamics Group, Los Alamos National Laboratory, Tech. Rep. LACC-98-16 v.3.

Lipscomb, W. H., 2001: Remapping the thickness distribution in sea ice models. *J. Geophys. Res.*, **106**, 13,989–14,000.

Rothrock, D. A., 1975: The energetics of the plastic deformation of pack ice by ridging. *J. Geophys. Res.*, **80**, 4514–4519.