

2011-04

Using CCSM3 for Paleoclimate Applications

Nan Rosenbloom
Christine Shields

Esther Brady
Sam Levis
Stephen Yeager

Climate Change Research
Climate and Global Dynamics Division

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

P. O. Box 3000
BOULDER, COLORADO 80307-3000
ISSN Print Edition 2153-2397
ISSN Electronic Edition 2153-2400



NCAR TECHNICAL NOTES

<http://www.ucar.edu/library/collections/technotes/technotes.jsp>

The Technical Notes series provides an outlet for a variety of NCAR Manuscripts that contribute in specialized ways to the body of scientific knowledge but that are not suitable for journal, monograph, or book publication. Reports in this series are issued by the NCAR scientific divisions. Designation symbols for the series include:

EDD – Engineering, Design, or Development Reports
Equipment descriptions, test results, instrumentation,
and operating and maintenance manuals.

IA – Instructional Aids
Instruction manuals, bibliographies, film supplements,
and other research or instructional aids.

PPR – Program Progress Reports
Field program reports, interim and working reports,
survey reports, and plans for experiments.

PROC – Proceedings
Documentation or symposia, colloquia, conferences,
workshops, and lectures. (Distribution may be limited to
attendees).

STR – Scientific and Technical Reports
Data compilations, theoretical and numerical
investigations, and experimental results.

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



Using CCSM3 for Paleoclimate Applications

Nan Rosenbloom

Climate Change Research

Christine Shields

Climate Change Research

Esther Brady

Climate Change Research

Sam Levis

Terrestrial Sciences Section

Stephen Yeager

Oceanography

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
P. O. Box 3000
BOULDER, COLORADO 80307-3000



Table of Contents

Forward.....	8
Acknowledgements	8
1 Introduction.....	8
2 Offline Tools Overview	10
3 Near-Modern Paleoclimate Simulations.....	11
3.1 Land – Near-modern	11
3.1.1 Surface dataset.....	11
3.1.2 clm.buildnml_prestage.csh	12
3.1.3 Changing land ice.....	13
3.1.4 Changing landcover	14
3.1.5 PFT-physiology dataset	14
3.1.6 Runoff directional dataset (rdirc)	15
3.2 Atmosphere – Near-modern.....	16
3.2.1 Atmosphere Initial Condition	16
3.2.2 CAM3 Namelist options.....	16
3.3 Coupler - Orbital Year	17
3.4 Ocean – Near-modern.....	18
3.4.1 Raising Sea Level	19
3.4.2 Lowering Sea Level.....	19
4 Deep Time Paleoclimate.....	24
4.1 Required input – Deep Time	24
4.1.1 Input 1: topobathy.nc.....	25
4.1.2 Input 2: vegetation.nc	25
4.2 Atmosphere - Deep Time.....	26
4.2.1 Initial condition file creation	26
4.2.2 Other Atmosphere Forcing Files:	27
4.2.3 Namelist adjustments for physical forcing:	27
4.2.4 Other Atmosphere Issues	28
4.3 Land - Deep Time.....	29
4.3.1 Initial condition file: Arbitrary Initialization.....	29
4.3.2 Surface Data Forcing	29
4.3.3 PFT-physiology dataset	31
4.3.4 Runoff directional dataset.....	31
4.3.5 Other Land Issues	36
4.4 Ocean and sea ice - Deep Time	36
4.4.1 Designing your ocean grid	37
4.4.2 Bathymetry (KMT)	38
4.4.3 Ocean regions.....	39
4.4.4 Ice Initial Condition Details	40
4.4.5 Ocean Initial Condition Details.....	40
4.4.6 Building your ocean grid	42
4.4.7 Summary Tables for building the ocean grid	49
4.4.8 GOTCHAS (Potential Problems).....	49
5 Coupler Mapping.....	51
5.1 Component Model Mapping Files.....	52



5.1.1	Two-way atmosphere-ocean grid mapping	52
5.1.2	Correct mapping errors	53
5.1.3	Diagnostics for mapping files: scrip_test	53
5.1.4	Summary Tables for Component Mapping Files	53
5.2	Runoff Mapping File	54
5.2.1	Tool: mk_runoff_remap.csh	54
5.2.2	Tool: gen_runoffmap	55
5.3	Orbital forcing	56
6	CCSM3 Component Model Scripts and Runtime Issues	58
6.1	Script and Forcing Locale Philosophy	58
6.2	Atmosphere Script and Runtime Issues	59
6.2.1	CAM3 Datasets	59
6.2.2	CAM3 Namelist Parameters	59
6.3	CLM3 Script and Runtime Issues	60
6.3.1	CLM3 Datasets	60
6.3.2	CLM3 Namelist Parameters	61
6.3.3	More details on 'mksrf' versus surface-data	61
6.4	Ocean Script and Runtime Issues	62
6.4.1	POP Forcing Files	62
6.4.2	POP Initial condition file	62
6.4.3	POP Input templates and pop_in	63
6.4.4	Other Issues	64
6.5	Ice Script and Runtime Issues	65
6.5.1	CSIM Datasets	65
6.5.2	CSIM Initialization	66
6.5.3	CSIM Namelist Parameters	66
6.6	Coupler Script and Runtime Issues	66
6.6.1	CPL6 Datasets	66
6.6.2	CPL6 Namelist Parameters	67
6.6.3	Other Issues	67
7	Diagnostics	68
7.1	Log files	68
7.2	Dagnostic flags in env_run	68
7.3	A few words about post-processing	68
8	Appendix	69
8.1	Setup Tools	69
8.2	Summary Table for Initial Conditions and Forcing Files	69
8.3	Useful links	70
8.3.1	Component Model User Guides:	70
8.3.2	Web sites useful for processing and viewing data files:	70
8.4	QuickGuide: CCSM3.0 Setup	71
8.5	LSM vegetation types	73
8.6	CLM3 PFTs	74
8.7	QuickGuide: kmtEd – Deep Time	74
8.8	Startup near-modern glacial simulation	76
8.9	Hybrid near-modern glacial simulation	77
8.10	Modifying the present day POP2 KMT	78
8.11	Coupler Mapping	79
8.12	River runoff mapping to the ocean grid	80
8.13	References	80



List of Figures

Figure 1: Paleo CCSM3 Setup	9
Figure 2: Changing land/ocean mask	20
Figure 3: Sea level lowstand on the ocean grid.....	22
Figure 4. Deep Time Flowchart.....	24
Figure 5. Height about sea level (m) for the late Cretaceous (80 Ma).....	25
Figure 6. RTM directional routing key	31
Figure 7. River Runoff.....	35
Figure 8. Ocean Grid using kmtEd.	40
Figure 9: Startup glacial simulation	76
Figure 10: Hybrid glacial simulation	77
Figure 11: Changing the present day land/ocean mask.....	78
Figure 12: Coupler mapping.	79
Figure 13: Mapping river runoff to the ocean.....	80

List of Tables

Table 1: Required Software	10
Table 2: Optional software	10
Table 3: CLM3 forcing datasets	12
Table 4: Solar constant and trace gas namelist variables	17
Table 5: Required input	24
Table 6: Summary Table for cami_create_ccsm.csh:	27
Table 7: Solar constant and trace gas namelist variables	28
Table 8: Aerosol optical depth specification namelist variables.....	28
Table 9: Summary Table for paleo_mkraw.csh	30
Table 10: Summary Table for rdirc.csh.....	33
Table 11: Summary Table for check.csh	33
Table 12: Summary Table for plotdirc.csh.....	34
Table 13. Vertical grid file for KMT.....	38
Table 14: mk_grid.csh/ns_dipole.f90 details	43
Table 15: mk_grid.csh/paleotopo.f90 details.....	44
Table 16: mk_grid.csh/grid_bin2nc.F90 details.....	45
Table 17: mk_ocninput.csh/input templates.....	48
Table 18: Summary Table for mk_grid.csh	49
Table 19: Summary Table for kmtEd.....	49
Table 20: Summary Table for mk_ocninput.csh	49
Table 21: mk_remap.csh details	52
Table 22: Summary Table for mk_remap.csh	53
Table 23: Summary Table for correct_map_error.run	54
Table 24: Summary Table for scrip_test.csh	54
Table 25: mk_runoff_remap.csh details.....	54
Table 26: gen_runoffmap namelist	56



Table 27: Summary Table for mk_runoff_remap.csh	56
Table 28: Summary Table for gen_runoffmap	56
Table 29: orbital parameters for > 1Ma	57
Table 30: Cam forcing datasets.....	59
Table 31: CLM3 forcing datasets	60
Table 32: Ocn forcing datasets changes for paleo	62
Table 33: Summary Table for Initial Conditions and Forcing Files	69



Forward

The Community Climate System Model (CCSM) and its component models have been invaluable tools for the national and international research communities. These models have been used by researchers worldwide, starting with the first version CSM1, to explore problems of both greenhouse and icehouse past climates. They have been shown to be adaptable for both deep-time paleoclimate applications, when the continental configurations were much different from present, to the more near-term past climates of the glacial-interglacial cycles of the last million years. Over 60 research articles using versions of CCSM for studying past climates have been published by the community. Low-resolution versions of CCSM have been developed and supported to allow paleoclimate researchers to integrate the model for thousands and even tens of thousands of years. This technical note outlines the procedures for setting up paleoclimate simulations with CCSM3, the version of CCSM released in 2004. The science of the paleoclimate simulation still rests on the researcher. Choices will need to be made on the appropriate conditions (i.e., land-ocean configuration, topography and bathymetry, ice sheets, vegetation) for the specific paleoclimate research topic being explored. Additional information on the component models and options that may need to be set can be found in the technical notes of these models. We thank all those who have contributed their time and expertise for maintaining the paleoclimate versions of CCSM.

Bette Otto-Bliesner and Jeff Kiehl

Acknowledgements

NCAR is sponsored by the National Science Foundation. C. A. Shield's contribution was supported by the Sedimentary Geology and Paleobiology (SGP) Program within the National Science Foundation. We thank our reviewers for their comments and we are grateful to the paleoclimate community for their many insights and suggestions over the years.

1 Introduction

This document describes the procedures for creating a CCSM3 paleoclimate simulation in the fully coupled (all active components) configuration. We provide tools and examples of the process used to create paleoclimate simulations using the computing resources at the National Center for Atmospheric Research (NCAR). This document is to be used as a guide; researchers are ultimately responsible for modifying the process to accommodate their time period of interest as well as adapting the tools to their available computer resources.

Throughout this User's Guide we differentiate between the procedures required to create (1) near-modern (e.g., Quaternary, Pliocene) or (2) Deep-Time (pre-Quaternary) model



simulations. In near-modern simulations, the continents are in their present-day positions, and the land/sea masks do not require significant modification. Quaternary modelers are often able to use existing forcing files to simulate past climate. By contrast, deep-time simulations require drastic modifications to the land/sea mask, and the modeler is responsible for providing the orographic/bathymetric maps for their geologic period of interest.

This document assumes a default fully coupled CCSM3 configuration. We do not describe the creation of the forcing files used in Data Model components or in stand-alone component model runs. To gain more understanding about the CCSM3 component models and input files, see the CCSM3 documentation (<http://www.cesm.ucar.edu/models/ccsm3.0/>), and refer to the Appendices in this document for links to CCSM3 User guides and useful model set-up tools.

The CCSM3 paleoclimate liaisons are available for consultation. Please see the CCSM3 webpage for contact details. (http://www.cesm.ucar.edu/working_groups/Paleo).

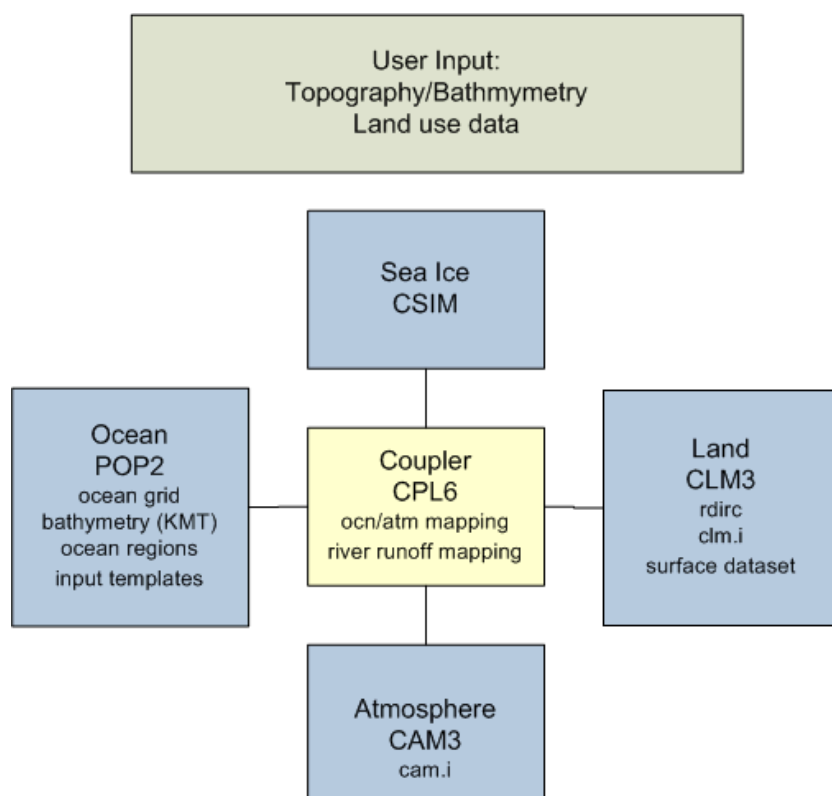


Figure 1: Paleo CCSM3 Setup

Schematic of PaleoCCSM3 initial and forcing files. The modeler must provide topography, bathymetry, and land cover files. The forcing files required to run CCSM3 can be created from the user provided input files using setup tools and guidance outlined in this document.



2 Offline Tools Overview

Table 1: Required Software

Required Software	Description
NCO	The netCDF Operators, or NCO, are a suite of programs known as operators . Each operator is a standalone, command line program which is executed at the UNIX shell-level. The operators take netCDF files as input, then perform a set of operations (e.g., deriving new data, averaging, hyperslabbing, or metadata manipulation) and produce a netCDF file as output. The operators are primarily designed to aid manipulation and analysis of gridded scientific data.
ncview	Ncview is a visual browser for netCDF format files. Ncview is not an analysis package; its purpose in life is to quickly and easily view simple plots of data stored in netCDF format.
NCL	The NCAR Command Language (NCL), a product of the Computational & Information Systems Laboratory at the National Center for Atmospheric Research (NCAR) and sponsored by the National Science Foundation , is a free interpreted language designed specifically for scientific data processing and visualization.
scrip1.4	Data interpolation software developed at Los Alamos National Laboratory. Available as a tarball from NCAR upon request. No support available.
kmtEd	GUI software designed to hand-edit points on a sphere.
Fortran90	Scientific computing language, Fortran 90 (or Fortran 77)

Table 2: Optional software

Optional software	Description
IDL	IDL® is a commercial software package for data analysis and visualization.
Matlab	MATLAB® is a commercial software package for data analysis and visualization. It is a high-level language and interactive environment.



3 Near-Modern Paleoclimate Simulations

In Quaternary simulations, the continents are in their present-day positions, and the land/sea masks do not require significant modification. Therefore, Quaternary modelers are often able to use existing forcing files and initial files. We describe a common procedure for adding continental ice sheets over North America and for lowering global sea level.

3.1 Land – Near-modern

The land model in CCSM3 is the [Community Land Model 3.0](#) (CLM3). If you plan to make changes to the land surface (topography, land cover, land ice, etc.) you will need netCDF files with your input topography at 10min and 1° resolution, and your land cover (vegetation) at 0.5° resolution.

3.1.1 Surface dataset

Some near-modern simulations are able to use the present day (default) surface dataset pointed to in the build script (clm.bldnml_prestage.csh) and Section 6.3:

```
e.g., set fsurdat = 'surface-data.128x064_atm.gx1v3_ocn.080101.nc'
```

However, if you have made changes to your land cover (e.g., by adding/removing land ice) or by changing vegetation, you will first need to change the raw input datasets to reflect these changes, and then create a new surface dataset.

CLM3 creates a new surface dataset (e.g., surface_data_048x096.nc) at runtime from seven mksrf files that contain 'raw' data on 0.5° (e.g., mksrf_glacier.nc) or 1° lat/lon grids (e.g., mksrf_lanwat.nc). The raw data include high-resolution maps of land cover PFTs (plant functional types), soil color, soil texture, leaf/stem areas and heights (LAI), land water (lakes and wetlands), and glaciers. Urban areas are set to zero in CCSM3 (Table 31). At model runtime, CLM3 reads in the mksrf files and creates a new surface dataset in the run directory, under the land subdirectory.

Hint: Change the generic name of the new surface dataset to something more descriptive by adding your case ID and date and placing the surface dataset in your \$CASEROOT/SourceMods/src.clm/ directory for future reference. Then point to your new surface dataset in clm.buildnml_prestage.csh:

```
Example: set fsurdat = 'surface_data_LGM_64x128.091012.nc'
```

Quaternary modelers can often use the default CLM3 mksrf files. However, some time periods may require modifications to the present day mksrf files to simulate land cover or vegetation change.



If your simulation requires drastic revisions to the 'raw' mksrf files, Deep Time Section 4.3.2.1 describes a tool for deep time experiments where continental configurations are very different from modern, and/or little is known about land cover and soil distributions.

NOTE: If you have lowered sea level and exposed new land along the continental shelf, CLM3 will automatically define that new land as 'wetland'. To reassign these points to another PFT you will need to modify your mksrf_pft file to fill in the new land points. You will also need to define LAI, SAI, MONTHLY_LAI, MONTHLY_SAI, MONTHLY_HEIGHT_TOP, MONTHLY_HEIGHT_BOT, and soil color for these cells. We recommend using a nearest neighbor algorithm to assign values for new land cells.

NOTE: Modifying mksrf_pft.nc can be tricky and drastically changing vegetation may result in a climate signal that is larger than the forcing (e.g., solar) that you are trying to simulate. Therefore, some Quaternary modelers choose to use present day land cover so that they can compare directly with present day or pre-industrial control simulations.

3.1.2 clm.buildnml_prestage.csh

Table 3: CLM3 forcing datasets

Namelist variable	filename	Type	Description/filename
fsurdat ¹		netCDF	Surface data file
finidat ²		netCDF	CLM3 initial condition file
fpftcon	pft-physiology	ASCII	PFT physiology specifications
frivinp_rtm ⁴	rdirc.05	ASCII	River transport model initial file
mksrf_fsoicol ⁵	mksrf_soicol_clm2.nc	netCDF	soil color
mksrf_flanwat	mksrf_lanwat.nc	netCDF	land water (lakes, wetlands)
mksrf_fglacier	mksrf_glacier.nc	netCDF	glacier
mksrf_furban	mksrf_urban.nc	netCDF	urban
mksrf_flai	mksrf_lai.nc	netCDF	leaf area index (LAI)
mksrf_fvegtyp	mksrf_pft.nc	netCDF	plant function types (PFTs)
mksrf_fsoitex	mksrf_soitex.10level.nc	netCDF	soil texture

¹The surface-data set is created at runtime using the raw (mksrf) datasets if fsurdat = '. Once the surface-data file is created, you can point directly to your new surface dataset.

Example: fsurdat = 'surface-data_64x128_8.5ka.101011.nc'

²The finidat file contains spun-up CLM3 state variables file. The finidat file must use the same PFT assignments and land/ocean mask as your simulation. Set finidat= '' for arbitrary initialization.



³ Unless the pft-physiology file was modified, the default file/location may be used. Otherwise, specify the unique filename and location.

⁴ The default name for the RTM initial condition file is rdirc.05. Near modern simulations may use the default rdirc file even if they have added land ice to North America.

⁵ The mksrf (raw) datasets contain land use information on a 0.5° grid. When the model is configured, the configure script automatically points to present day mksrf files in the CLM3 namelist. If you are using modified mksrf files generated by paleo_mkraw or convert_mksrf, you must point to these new mksrf files in clm.buildnml_prestage.csh. Note that pointing to these files in the namelist DOES NOT guarantee their use. These files are only used if fsurdat= ``. See 6.3.3 for further information.

3.1.3 Changing land ice

If your experiment requires a change in land ice, we provide a tool called convert_mksrf.F90 that will modify the default mksrf files to reflect changes in glaciers, lakes/wetlands and PFTs, and produce new mksrf_glacier_myrun.nc, mksrf_lanwat_myrun.nc and mksrf_pft_myrun.nc files. The program requires a netcdf file containing your surface variables at 10min resolution: topo-ice.10min.nc

```
float TOP(lat, lon) ;
    TOP:units = "meter" ;
    TOP:long_name = "10-min elevation from USGS 30-sec dataset" ;
float lat(lat) ;
    lat:long_name = "lat" ;
    lat:units = "degrees north" ;
float lon(lon) ;
    lon:units = "degrees east" ;
    lon:long_name = "lon" ;
float ICE(lat, lon) ;
    ICE:units = "%" ;
    ICE:long_name = "Fraction of Grid Cell Covered with Glacier" ;
    ICE:valid_min = 0.f ;
    ICE:valid_max = 100.f ;
float LANDFRAC(lat, lon) ;
    LANDFRAC:valid_max = 1 ;
    LANDFRAC:valid_min = 0 ;
float LANDMASK(lat, lon) ;
    LANDMASK:valid_max = 1 ;
    LANDMASK:valid_min = 0 ;
```

3.1.3.1 Convert_mksrf.F90

- Tool: convert_mksrf.F90, convert_mksrf.template
- Input: topo-ice.10min.nc
- Output:
 - mksrf_glacier_myrun.nc
 - mksrf_lanwat_myrun.nc
 - mksrf_pft_myrun.nc



Usage:

- Step 1. `cp convert_mksrf.template convert_mksrf.template.myrun`
- Step 2. modify `convert_mksrf.template.myrun` to point to your input files
- Step 3. `cp convert_mksrf.template.myrun convert_mksrf.F90`
- Step 4. compile (gmake)
- Step 5. execute: `./convert_mksrf`

3.1.4 Changing landcover

CLM3 does not require an initial condition file and can be initialized with arbitrary initialization (Section 6.3.1) and the surface dataset created from your mksrf datafiles.

You can also branch from an existing simulation, using `clm2.i` or `clm2.r` (restart) files to initialize your new experiment. `Clm.i` files are often desirable because they contain spunup data (e.g., carbon pools) from an existing run.

However, if you have changed land cover in your new simulation (for example, if you have changed your land ice distribution), you will first need to create a new `clm2.i` file that conforms to your new land cover assignments. Creating a new `clm2.i` file is a two step process:

1. run a 5 day startup simulation with arbitrary initialization (`finidat = ```)
2. re-map the land properties from the old `clm2.i` file, onto your new `clm2.i` file using `interpinic`.
3. Use the surface dataset created by the 5 day run for your new experiment.

The new `clm2.i` file produced by the 5-day startup run has `landunit` vector lengths consistent with the new surface dataset, but different from the original experiment that you wish to branch from. Use the tool, `interpinic`, to project the spun-up data from the initial `clm2.i` file onto the new initial file created by the 5-day startup run.

3.1.4.1 Interpinic

- Tool: `interpinic`
- Source: `CCSM3_source/models/land/clm2/tools/interpinic`

`Interpinic` maps land use data from one `clm` initial file (the input file) to another `clm` initial file (the output file) by overwriting the contents of the output file. The input and output files may be of any spatial resolution and `gridcell/landunit/column/pft` configuration.

Usage: `interpinic -i old.clm2.i.nc -o new.clm2.i.nc`

3.1.5 PFT-physiology dataset

The CLM3 land model defines the physiology of each plant functional type (PFT) in an ASCII text file, called 'pft-physiology'. The default pft-physiology definitions are generally used for paleo experiments. However, if you wish to change the characteristic of a specific CLM3 PFT you may need to edit this dataset. Please read the [CLM3 documentation](#) (Section 8.3.1) before altering this file.



3.1.6 Runoff directional dataset (rdirc)

The River Transport Model (RTM) runs inside the land model on a fixed regular grid that is different from the parent CLM3 grid (the CLM3 and CAM3 models use the same grid, which for CCSM3, is a Gaussian grid). The runoff directional dataset required for RTM is an ASCII file containing latitude, longitude, and an integer value describing the vector (direction) for runoff flow at each RTM grid point. The integer values are numbered from 1 to 8: 1=N, 2=NE (45°), 3=E (90°), etc. See Section 4.3.4 for complete vector directions and integer labels.

Near-modern experiments are usually able to use the default runoff directional dataset (rdirc.05) pointed to in `clm.buildnml_prestage.csh`.

3.1.7 Other Land Issues

Model dynamics parameters tend to be resolution dependent and can be controlled in the model namelist. These parameters may not need to be altered from the default CCSM3 configuration, but if they do, instructions will be covered in Chapter 5, *Model Setup Scripts and Runtime Issues*.

There are several source code modifications the modeler may need to make.

1. mksrfdatMod.F90

1.1 In the default CLM3 source code, the Ross Ice Shelf is hardwired into the `mksrfdatMod.F90` code. If you wish to remove the Ross Ice Shelf, copy `mksrfdatMod.F90` to your `SourceMods/src.clm` directory and search for 'Ross Ice Shelf'. Comment out the entire section referring to setting LAND values on the Ross Ice Shelf to glacier.

1.2 This modification is recommended for all users.

Original:

```
pctwet(i,j) = 100. - pctlak(i,j)
pctgla(i,j) = 0.
```

Modified:

```
pctwet(i,j) = 100. - pctlak(i,j) - pctgla(i,j)
! pctgla(i,j) = 0.
```

1.3 Additionally, if you have lowered sea level and exposed new land along the continental shelf, CLM will automatically define that new land as 'wetland'. If you desire another vegetation designation for these newly created land areas you can modify `mksrfdatMod.F90` to fill in these areas (e.g., with nearest neighbor land cover).

2. clm_varpar.F90



If your RTM forcing data set (produced by `rdirc.csh`) is NOT 0.5 degrees, the modeler will need to edit this code to specify the correct latitudes and longitudes. Search for `rtmlon` and `rtmlat`.

All code modifications need to be placed in the `$CASEROOT/SourceMods/src.clm` subdirectory of your CCSM3 case. See Appendix 8.4 and CCSM3 user documentation for details on how to create a new case and run CCSM3.

3.2 Atmosphere – Near-modern

The atmospheric model in CCSM3 is the [Community Atmosphere Model 3.0](#) (CAM3).

3.2.1 Atmosphere Initial Condition

- Tool: `definesurf`
- Source: `setup_tools.tar`

If you change the land surface elevation from present day (e.g., increasing the height of the Greenland Ice Sheet, or lowering the Rocky Mountains) you must create a new topography file for the atmosphere model (CAM3) that reflects these changes.

Surface topography in CCSM3 is parameterized by the CAM3 variable PHIS (surface geopotential [m^2/s^2]), which is defined by the relationship:

$$\begin{aligned}\text{PHIS } [\text{m}^2/\text{s}^2] &= \text{elevation}[\text{m}] \times \text{gravity } [\text{m}/\text{s}^2] \\ \text{PHIS} \div 9.81 \text{ m}/\text{s}^2 &= \text{elevation } [\text{m}]\end{aligned}$$

`Definesurf` is a fortran-based tool designed to create a new topography input file. Near-modern paleo modelers can use `definesurf` by providing a netCDF file of their paleo topography at 10min resolution.

Usage: `definesurf -t topofile.10min.nc -g gridfile -l landm_coslat.nc outfile`

Users are advised to create their new topography file by first creating an anomaly map of their time period relative to present day ($\Delta Z = \text{topo}_{\text{my_TOPO_paleo}} - \text{topo}_{\text{my_TOPO_present-day}}$) and then adding this anomaly (ΔZ) to the CCSM3 present-day base topography (`USGS-gtopo30_10min_c050419.nc`; distributed in `setup_tools.tar`).

If atmospheric initial conditions are completely unknown we provide an ncl tool (`cam_create_ccsm3.csh`) that will create an initial file (CAMI) that will reflect a basic physical atmospheric state suitable for initialization that allows the atmospheric model to spin-up during the first few decades of integration. This tool is primarily used for Deep-Time modeling, and is described in 4.2.1.

3.2.2 CAM3 Namelist options

More details on `cam.buildnml_prestage.csh` are discussed in Section 6.2.2. This section describes the namelist parameters that control aspects of your physical boundary forcing.



A: Solar constant and trace gases

You must assign appropriate values for the solar constant and for trace gas concentrations.

Note: Orbital parameters (as opposed to the solar constant) are set in the coupler namelist for a fully coupled CCSM3 experiment, and not in the CAM3 namelist. This is different from CAM stand-alone or CAM-SOM runs.

Table 4: Solar constant and trace gas namelist variables

VARIABLE	Description
SCON	Solar constant
CO2VMR	CO ₂ volume mixing ratio
CH4VMR	CH ₄ volume mixing ratio
N2OVMR	N ₂ O volume mixing ratio
F11VMR [†]	CFC11 Volume mixing ratio
F12VMR [†]	CFC12 volume mixing ratio

[†]For Pre-Industrial paleo experiments, F11VMR and F12VMR should be set to 0.

B: Atmospheric Forcing Files

- **Absorption/Emissivity: [absdata]**
- The present day absorption/emissivity forcing dataset was built with wide constraints and is therefore flexible and can be used for paleoclimate cases.
 - abs_ems_factors_fastvx.c030508.nc [CCSM3.0 Release DEFAULT]
- **Ozone: [bndtvo]**
- Typically, present day or pre-industrial ozone mixing ratio boundary forcing files are used for paleoclimate cases. Ozone forcing files are independent of resolution.
 - pcmdio3.r8.64x1_L60_clim_c970515.nc [CCSM3.0 Release DEFAULT]
 - moztart.o3.128x64_L18_1870cyc_c040123.nc [Pre-Industrial]
- **Aerosols: [aeroptics][bndtvaer]**
- Pre-industrial or present day aerosol mass files are resolution dependent. We provide T31 and T42. Please contact the Paleo Liaison if you need a different resolution.
 - AerosolOptics_c040105.nc [CCSM3.0 DEFAULT]
 - AerosolMass_V_64x128_clim_c031022.nc [CCSM3.0 DEFAULT]
 - AerosolMass_V_1870_SO4_64x128_clim_c040224.nc [Pre-Industrial]

3.3 Coupler - Orbital Year

To simulate changes in solar irradiance through time, the model calculates the eccentricity, obliquity and precession based on Berger et al., 1993. The model sets these parameters automatically using a variable called orb_year, which is set in the coupler namelist (cpl.buildnml_prestage.csh) and is defined in years before 1950.

a. AD values, i.e. 1800, 1900, 1950, 1990, 2000, etc should be expressed with the explicitly named year.



example: orb_year = 1800

b. BC values, i.e. 10ka, 130ka, 506ka, should be expressed in years relative to 1950 (1950 - <time period of interest>)

examples:

- o 506ka BP: orb_year = (1950 - 506ka) = -504050
- o 4ka BP: orb_year = (1950 - 4000) = -2050

Since we have no good estimate of the orbital variations beyond ~1Ma BP, eccentricity, obliquity, and the moving vernal equinox must be expressed individually for pre-Quaternary experiments. Details on how these variables are computed can be found by reviewing the code in the CCSM3 csm_share subdirectory.

3.4 Ocean – Near-modern

There are many reasons why paleo-climate modelers may be interested in changing the land/sea mask. In this section we discuss changes to the modern land/sea mask to simulate sea level low stands at the Last Glacial Maximum (LGM), and extending continental glaciers across North America. Examples of other simulations that would require this process include:

- Changing sea level
- adding land ice over North America, covering Hudson Bay
- open/closing straits (e.g., Bering Strait, Isthmus of Panama)
- removing Hudson Bay for pre-Pleistocene simulations

The process of altering the modern land/ocean mask is a complicated, multistep process (Figure 2, Figure 11), and we offer these general guidelines:

- Raising or lowering sea level requires changing the land/sea mask along the continental margins and creating new coupler masking files. Raising sea level requires that land cells be re-defined as ocean. Lowering sea level requires that ocean cells be re-defined as land cells. In general, changing cells from ocean to land is more straightforward than turning land into ocean.
- We do not recommend changing ocean bathymetry/seafloor relief (KMT) when including changes in sea level (other than to make localized changes that are essential to your science question: e.g., closing the Bering Strait, or opening the Isthmus of Panama). Model results will be very sensitive to ocean depth (KMT), biasing the climate signal you may be interested in. If you change ocean depth you should also run a control experiment with your altered KMT to test the model sensitivity to your new KMT.

When you change the land/sea mask (i.e., re-assigning ocean cells as land cells) you need to create a new binary KMT file. The tool change_kmt.ncl provides a template for making changes to the land/sea mask and producing a new binary KMT file that will be



read by the ocean model, and used to create new coupler mapping files. For an overview of the steps required to set up a near-modern experiment where sea level has been lowered, or the ocean/land mask has been altered see Appendix 8.8 and 8.9.

3.4.1 Raising Sea Level

Re-assigning land to ocean is a complicated process because it requires setting land points to ocean, which then must be initialized. The ocean requires that all points be initialized at start up. If you configure your experiment as a startup run, any newly ocean cells will automatically be filled with Levitus (Levitus et al., 1998, Steele et al., 2001) values. However, it is generally desirable to initialize the ocean from a branch or hybrid control experiment, because of the long time scales required to spin up the deep ocean. In order to use a previous ocean restart file, you will need to modify the ocean source code to initialize the temperature and salinity fields at all KMT levels for the newly created ocean cells. This process can be complex and is beyond the scope of this document.

3.4.2 Lowering Sea Level

Changing ocean points to land is more straightforward than removing land points, but it still requires multiple steps (Figure 2). **Step 1:** changing the ocean bathymetry (KMT) file. **Step 2:** remaking the coupler mapping files that drive communication within CCSM3 (See Chapter 5). **Step 3:** creating a new initialization (CAMI) file for CAM3. **Step 4:** creating new 0.5° resolution raw 'mksrf' datafiles that will be used by CLM3 at runtime to create a new surface dataset consistent with your land configuration. In contrast to the case of simulated sea level rise, sea level lowering simulations can generally restart the ocean and sea ice models using restart files from present-day simulations, as long as no new ocean points have been added. **Steps 4a-4b:** are required only if you are restarting from a previous case, but have changed land cover (e.g., added land ice, changed vegetation distributions, or changed the land/ocean mask).



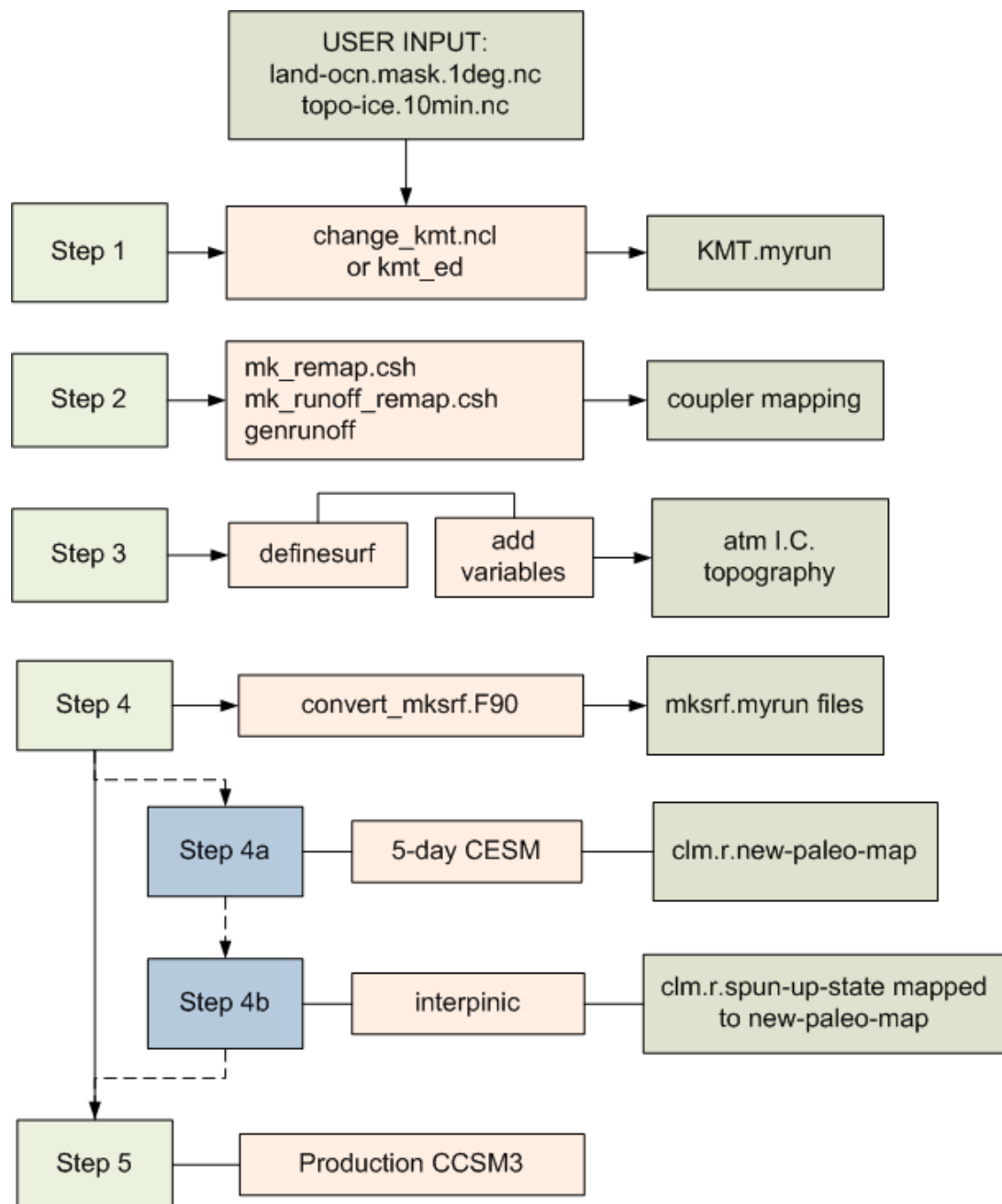


Figure 2: Changing land/ocean mask

Schematic outline of the steps required to change the land/ocean mask and the ocean bathymetry (KMT) file.

Step 1: Modify binary ocean KMT file

- Tool: `change_kmt.ncl`
- Input: 1 degree land/ocean mask (netCDF)



- Source: setup_tools.tar

The NCL script `change_kmt.ncl` remaps ocean points to land to include changes in sea level (Figure 11). It may be used in a very simple way to remove the ocean cells representing Hudson Bay. Or you may use your own topography input file to change ocean points to land globally (e.g., to extend the continental margins for sea level lowstands) (Figure 3). The NCL code requires the default binary CCSM3 topography (KMT) and region mask files as input, and produces two new binary files with the user-defined changes to the ocean map. The new binary files are used by the ocean model (`pop.buildnml_prestage.csh`) and are input to the paleo tool, `mk_remap.csh`, to create new coupler mapping files that map the new land/ocean map to the atmosphere (Figure 12). Note that the new binary files must be written in big-Endian binary format.

Hint: Check the region mask to be sure you have not eliminated any ocean regions. Eliminating ocean regions requires renumbering the ocean region in the region mask file and changing the `region_ids` file (e.g., `gx1v3_region_ids`) correspondingly. By retaining at least one active ocean cell in the default regions, you can avoid having to change the `region_ids`.

Hint: The NCL code we provide is a template for making macro changes to the default binary KMT map. For best ocean model results, carefully examine your new KMT ocean/land mask for newly emergent islands in the Pacific, opening closed basins, evaluate changes to your marginal seas, and widen or eliminate narrow channels. These changes will require hand-editing of the `change_kmt.ncl` code to fix these problems. We recommend using `ncview` on the netCDF output file from `change_kmt.ncl` to identify cells you would like to modify (Figure 3). See Section 4.4 for ocean grid and KMT overview and recommendations.



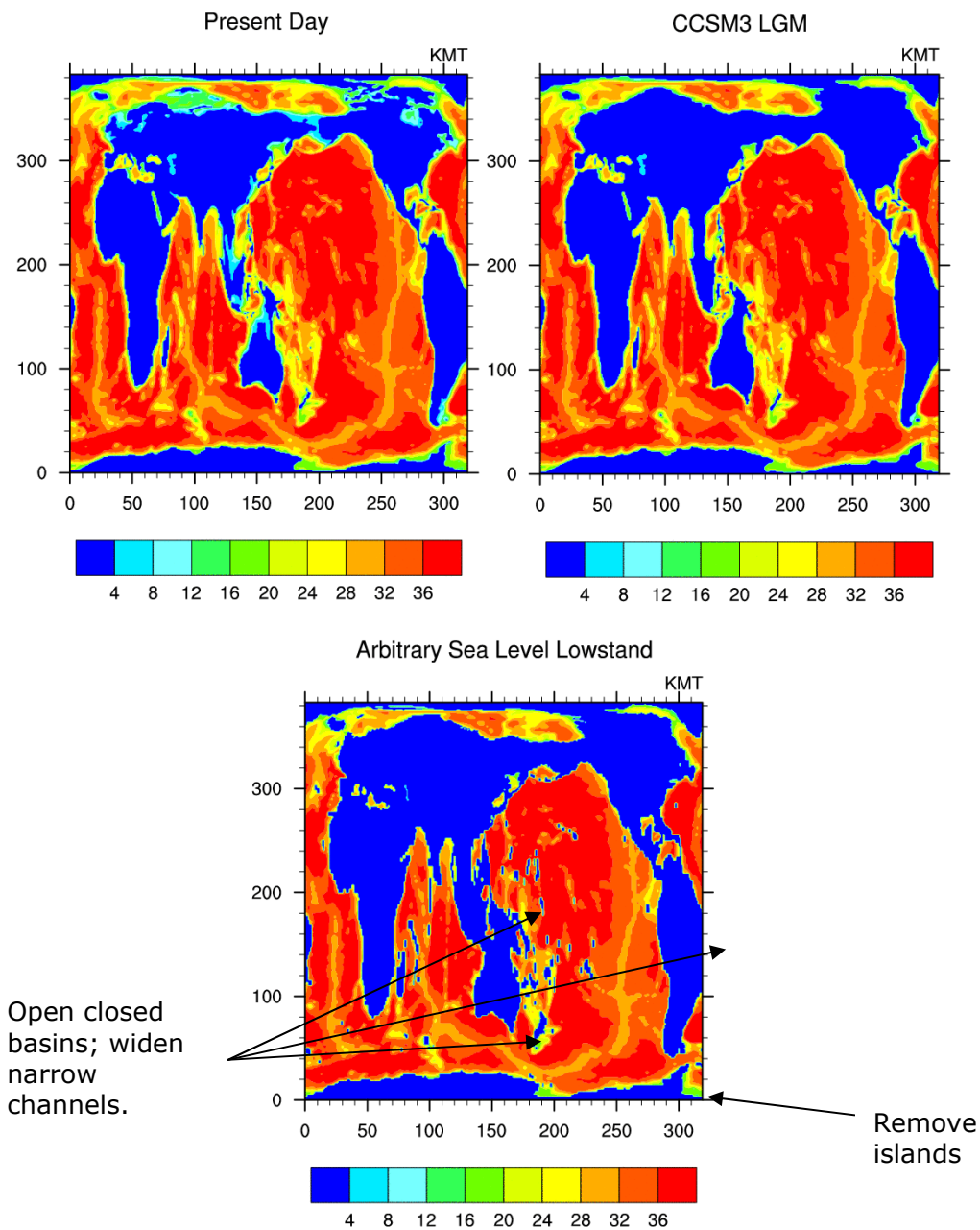


Figure 3: Sea level lowstand on the ocean grid.

Lowering sea level to expose the continental shelf during a sea level low stand requires changing the land/sea mask by modifying the ocean bathymetry (KMT) file. Be sure to open closed basins, and widen or eliminate narrow channels; for better model results remove mid-Pacific islands.

Step 2: Remap coupler files (Section 5)



Step 3: Create new cam.i file (Section 3.2)

Step 4: Create new clm.i file (Section 3.1)

Steps 4a–4b: 5-day simulation for hybrid start (Section 3.1.4.1)

If you have changed the land/ocean mask, but would like to restart the land model with the spunup carbon pools from a previous CCSM3 simulation, you will need to run a 5-day CCSM3 coupled startup simulation to create a new surface dataset, and a clm.i file that will then be used as a template for remapping the spun-up CLM pools onto your new run.



4 Deep Time Paleoclimate

This chapter will describe how to create the necessary forcing and initial condition files for each model component for deep time paleo cases. (See Summary Table: Appendix 8.2). Additionally, a pre-staging setup script is available to create a convenient directory structure for all of the setup tools. See Appendix 8.1 for a description of the setup tools described in this document.

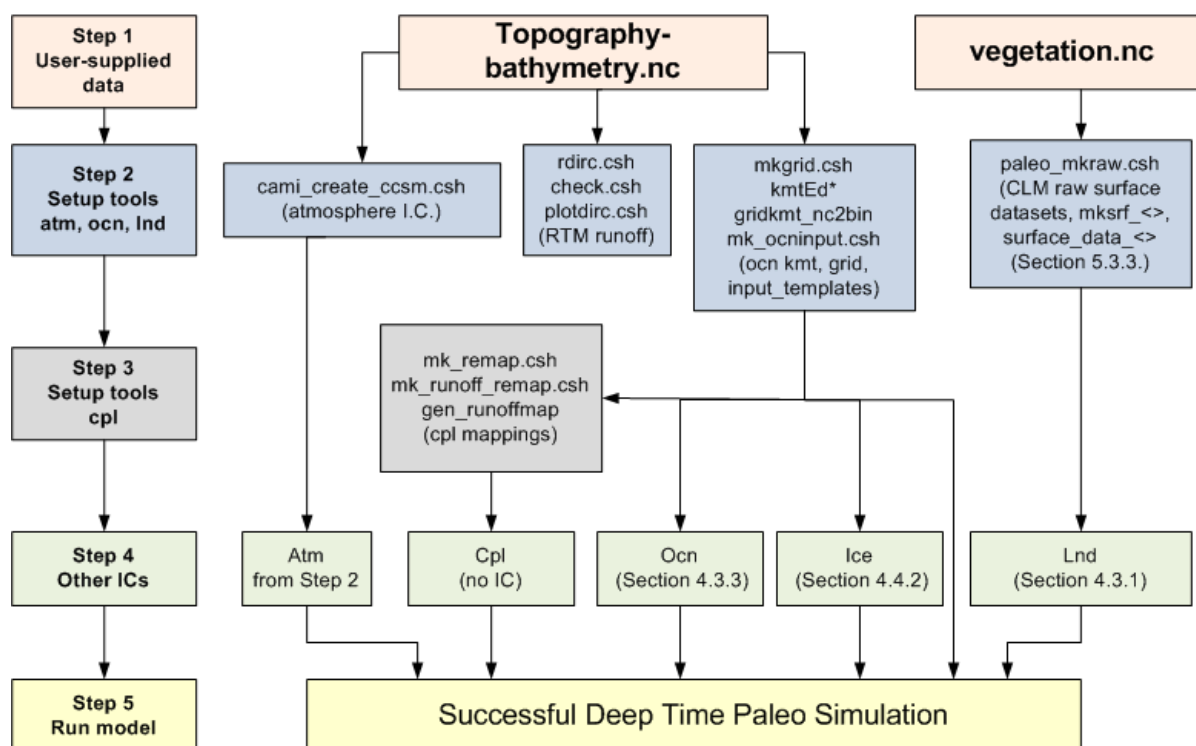


Figure 4. Deep Time Flowchart.

4.1 Required input – Deep Time

To begin creating a deep time paleoclimate case, the researcher will need to provide two netCDF files with topography/bathymetry and land use (vegetation) on a regular latitude/longitude grid (e.g., $2^\circ \times 2^\circ$), using lat(lat) and lon(lon) as coordinate variables.

Table 5: Required input

Example Filename	Description	Notes
------------------	-------------	-------



topobathy.nc	Topography/bathymetry map on regular lat/lon grid.	NetCDF format. Coordinate vars: lat(lat), lon(lon) Longitude: 0 to 360 degrees Latitude: -89 to 89 degrees
vegetation.nc	Land use/vegetation on regular lat/lon grid.	NetCDF format. Coordinate vars: lat(lat), lon(lon) Longitude: 0 to 360 degrees Latitude: -89 to 89 degrees

4.1.1 Input 1: topobathy.nc

The topobathy.nc netCDF file should contain topography and bathymetry on a regular latitude/longitude grid (e.g., $2^\circ \times 2^\circ$). Positive values represent height above sea level and negative values represent ocean bathymetry. The topobathy.nc defines your land/ocean mask. An example of a topobathy.nc datafile is shown in Figure 5.

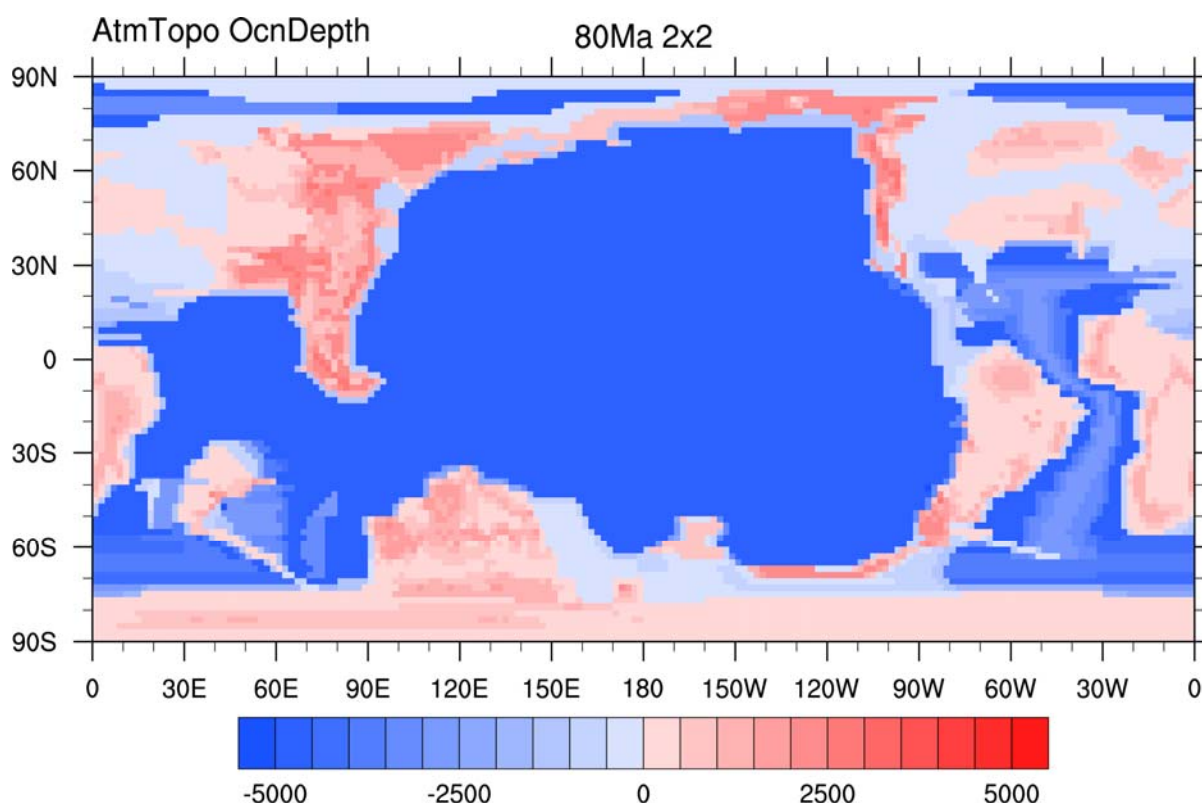


Figure 5. Height about sea level (m) for the late Cretaceous (80 Ma).
Data are at $2^\circ \times 2^\circ$ degree latitude/longitude resolution.

4.1.2 Input 2: vegetation.nc



The vegetation.nc netCDF file should contain land use (i.e., vegetation) on a regular latitude/longitude grid (e.g., $2^\circ \times 2^\circ$).

For most deep time paleo simulations, we assign LSM (Land Surface Model) land-use types (Appendix 8.5) for each grid point and then convert these LSM types to CLM3 (Community Land Model) surface information (Appendix 8.6) using the tool `paleo_mkraw.csh`. Because CLM3 requires a complicated array of surface information for each grid cell, whereas LSM uses a simple integer value to represent land-use at each grid point, assigning an LSM integer value and using the `paleo_mkraw.csh` tool to convert to LSM types to CLM3 format provides a simple method to create surface data information for deep time. The `paleo_mkraw.csh` tool is included in `setup_tools` and details on the script are discussed in Section 4.3.2.1.

HINT: Modelers may first need to construct LSM land cover maps from biome maps using the LSM definitions listed in the Appendix 8.5. The CLM3 PFTs definitions used in `paleo_mkraw.csh` are listed in Appendix 8.6. LSM land-use types are used in `paleo_mkraw.csh` because LSM was the predecessor to CLM3 and used in CSM1.4. If this does not suite your needs, you will need to modify `paleo_mkraw.csh` to convert from your preferred land-use type structure to CLM3 surface data information.

4.2 Atmosphere - Deep Time

Initial and boundary forcing files for the atmosphere model are listed in the Summary Table (Appendix 8.2). As a deep time modeler, you are responsible for creating a new initial condition file (`cam_i`) and using the CAM namelist functionality to specify trace gas constituents and the solar constant appropriate for your time period. Instruction on how to change the namelist parameters is discussed in Section 6.2.2. For further details on CCSM3/CAM scripts, please refer to the User Guides:

CAM3: <http://www.cesm.ucar.edu/models/atm-cam/docs/usersguide/>

CCSM3: <http://www.cesm.ucar.edu/models/ccsm3.0/ccsm/doc/UsersGuide/UsersGuide/UsersGuide.html>

4.2.1 Initial condition file creation

- Tools: `cam_i_create_ccsm.csh`
- `ccsm_cam_i_create_ccsm3.ncl`

We assume that for deep time periods, atmospheric initial conditions are unknown. Given this assumption, we attempt to provide a basic physical atmospheric state suitable for initialization that allows the atmospheric model to spin-up during the first few decades of integration. The tool used below computes a cosine weighted equator-pole latitudinal temperature distribution based on the user-specified polar and equator temperature estimates. Default values for these are currently set at 28°C (equator) and 12°C (pole). Surface pressure and the surface geopotential are computed based on the topography (`topobathy.nc`), and the zonal velocity is based on the thermal wind balance.



The c-shell script `cam_create_ccsm.csh` drives an NCL script called `ccsm_cam_create_ccsm3.ncl`. You will need your `topobathy.nc` file as well as a `cam` master (skeleton) file for your resolution. The scripts and master file are provided in `setup_tools.tar`. The master file is simply used as a placeholder for initial variables names and will be overwritten with the basic physical state computed by the NCL program. Therefore, if a master (skeleton) file for your resolution is not provided in `setup_tools.tar`, you can use any `cam` file from the CCSM3 distribution or from a previous CAM run.

You will need to edit the environmental variables in the script to values appropriate for your case. Details on this script can be found in the comments of the script itself.

Table 6: Summary Table for `cam_create_ccsm.csh`:

Shell Script	<code>cam_create_ccsm.csh</code>
Source code	<code>ccsm_cam_create_ccsm3.ncl</code>
Inputs	<code>topobathy.nc</code> <code>cam_skeleton_atm_resolution.nc</code>
Output	<code>cam_my case.nc</code>

4.2.2 Other Atmosphere Forcing Files:

- **Absorption/Emissivity:**
- The present day absorption/emissivity forcing dataset was built with wide constraints and is therefore flexible and can be used for paleoclimate cases.
 - `abs_ems_factors_fastvx.c030508.nc` [CCSM3.0 DEFAULT]
- **Ozone:**
- Typically, present day or pre-industrial ozone mixing ratio boundary forcing files are used for paleoclimate cases. Choice of dataset will depend on your control experiment.
 - `pcmdio3.r8.64x1_L60_clim_c970515.nc` [CCSM3.0 DEFAULT]
- **Aerosols:**
- Although it is necessary to input present day or pre-industrial aerosol boundary forcing files into the model, it is recommended that deep time paleoclimate researches use the `namelist` functionality to flag CAM to internally compute aerosol optical depths.
 - `AerosolOptics_c040105.nc` [CCSM3.0 DEFAULT]
 - `AerosolMass_V_48x96_clim_c031029.nc` [CCSM3.0 DEFAULT]

4.2.3 Namelist adjustments for physical forcing:



Details on where to specify the runtime namelist options is discussed in Section 6.2.2. This section describes only the namelist parameters that control aspects of your physical boundary forcing.

Group A: Solar constant and trace gases

You will need to determine appropriate values for the solar constant and for trace gas concentrations. Note that orbital parameters are set in the coupler for a fully coupled CCSM3 experiment, and not in CAM. This is different from CAM stand-alone or CAM-SOM runs.

Table 7: Solar constant and trace gas namelist variables

VARIABLE	Description
SCON	Solar constant
CO2VM2	CO ₂ volume mixing ratio
CH4VMR	CH ₄ volume mixing ratio
N2OVMR	N ₂ O volume mixing ratio
F11VMR [†]	CFC11 Volume mixing ratio
F12VMR [†]	CFC12 volume mixing ratio

[†]For deep time paleo simulations, F11VMR and F12VMR should be set to 0.

Group B: Aerosol Optical Depth

CAM radiation code requires a boundary forcing dataset for aerosol mass mixing ratios and aerosol optical properties. Because this is unknown for many paleoclimate cases, we must flag the code to use the CCM3.6 method for computing aerosol optical depths, which does not require spatial knowledge for aerosol mixing ratios. This method will compute a uniform optical depth across all grid points. Group B parameters must all be specified in the namelist.

A tauback value of 0.28 is equivalent to CCM3.6 default value of TAUVIS = 0.14. The various scaling parameters must be set to zero to render the present day aerosol mixing ratio values in the code obsolete and allow the uniform optical depth to be used.

Default values for all namelist parameters can be found in the [CAM3 User's Guide](#).

Table 8: Aerosol optical depth specification namelist variables

VARIABLE	Description	Deep time
TAUBACK	Uniform optical depth	0.28
CARSCL	Carbon scaling	0.0
DUSTSCL	Dust scaling	0.0
SSLTSC	Sea salt scaling	0.0
SULSCL	Sulfate scaling	0.0

4.2.4 Other Atmosphere Issues



Model dynamics parameters tend to be resolution dependent and can be controlled in the model namelist. These parameters may not need to be altered from the default CCSM3 configuration, but if they do, instructions will be covered in Chapter 5, *CCSM3 Component Model Scripts and Runtime Issues*.

4.3 Land - Deep Time

Initial and boundary forcing files for the land model are listed in the Summary Table (Appendix 8.2). Deep time paleo modelers are responsible for creating new surface data forcing as well as a directional runoff map on the River Transport Model (RTM) grid.

4.3.1 Initial condition file: Arbitrary Initialization

CLM3 does not require an initial condition file and can be initialized with arbitrary initialization (Section 6.3.1). Typically, for deep time, the researcher does not know the land model initial state with enough accuracy to improve upon the arbitrary initialization option. CLM3 will spin up its own variables at runtime. If the modeler does wish to use an initial condition file, refer back to Section 3.1 for ideas and strategies. An initial condition procedure for deep time is not supported.

4.3.2 Surface Data Forcing

Changing the land/sea mask for deep time model experiments requires that the modeler create a new land surface dataset. The surface dataset is constructed from a series of seven 'raw' datafiles, created by `paleo_mkraw.csh`. Instructions on modifying the CLM namelist to trigger the creation of a new surface dataset can be found in Section 6.3.

4.3.2.1 Tool: `paleo_mkraw.csh`

The script `paleo_mkraw.csh` uses the LSM vegetation types created by the user for your time period (`vegetation.nc`) and current day soil texture profiles (`mksrf_soitex.10level.nc`) to create the seven 'raw' land surface datafiles required by CCSM3. These datasets are named '`mksrf_[]`.nc' and include surface data information for PFTs (plant functional types), soil color, soil texture, leaf/stem areas and heights (LAI), land water (lakes and wetlands), glaciers, and urban areas (Table 9). From these 'raw' datasets, a single land surface data file, tailored to your land and ocean grids, will be created at model runtime (Section 4.3.2.2).

HINT: To create the `mksrf` files required for the surface dataset, you must specify the resolution of your incoming LSM vegetation dataset by setting longitude/latitude in `paleo_mkraw_sed.F90`. For example, if your LSM vegetation map is at 2x2 degree resolution, then `nlon=180` and `nlat=90`, and your `mksrf_[]`.nc files will also be 2x2



degree. If your vegetation file has 0.5 degree resolution, nlon=720 and nlat=360 and your mksrf_*.nc files will also be at 0.5 degree resolution.

The script paleo_mkraw.csh assumes that glaciers=urban=lakes=wetlands=0, soil texture =loam and soil color=4. If this does not suit your needs, you will need to alter paleo_mkraw_sed.F90 to make any desired changes.

For example, if you would like to specify glaciers, you will need to edit the subroutine create_mksrf_glacier and add code to test for LSM type 1 (land ice). For each point equal to 1, assign pct_glacier values from 0 to 100%. See section 3.1.3 for a discussion on adding glaciers to near-modern simulations.

Table 9: Summary Table for paleo_mkraw.csh

Shell script	paleo_mkraw.csh
Source Code	paleo_mkraw_sed.F90
Inputs	vegetation.nc mksrf_soitex.10level.nc
Output	mksrf_glacier_myrun.date.nc mksrf_pft_myrun.date.nc mksrf_lai_myrun.date.nc mksrf_soicol_myrun.date.nc mksrf_lanwat_myrun.date.nc mksrf_soitex_myrun.date.nc mksrf_urban_myrun.date.nc

4.3.2.2 Runtime surface_dataset

At model runtime, CLM reads in the new mksrf_[glacier,pft,lai,...]_myrun.nc datafiles and creates a new surface dataset file (surface_data_[resolution].nc) in the run directory, under the land subdirectory. Instructions on pointing to the raw mksrf files can be found in Section 6.3.3.

HINT: It is highly recommended that the modeler review the newly created surface_data_[resolution].nc for accuracy. When creating this dataset, CLM3/CCSM uses the ocean grid to compute the land/ocean mask and project the land variables onto atm/Ind grid. When this initial mapping is done, there are often mismatches between the land assignments made by the ocean grid and the Ind/atm grid. If the ocean model views a grid point as land, and the land model has no information for that grid point (because on the land/atmosphere grid it was considered ocean), then CLM3 assigns this point to be wetland. This typically occurs along the coastlines and may not be desired. The modeler will need to modify the surface_data_[resolution].nc file to correct for these erroneous wetland points. We provide an NCL script in the setup_tools tar file to correct surface_data_resolution.nc if necessary (ccsm_clm2_surface_create_pub.ncl).



HINT: You may want to change the generic name of the new surface dataset to something more descriptive by adding your case resolution, a descriptive case reference, and/or a date.

Example: surface_data_LGM_64x128.091012.nc

Hint: Once your surface_data_myrun.nc file is corrected, all model submissions should point to this file.

4.3.3 PFT-physiology dataset

The CLM3 land model defines the physiology of each plant functional type (PFT) in an ASCII text file, called pft-physiology. The default pft-physiology definitions are generally used for paleo experiments. However, if you wish to change the characteristic of a specific CLM PFT, and you cannot accomplish your goal by modifying paleo_mkraw_sed.F90, you may need to edit this dataset. Please read the CLM documentation before altering this file and/or contact a CCSM paleo liaison for a consultation.

4.3.4 Runoff directional dataset

Deep time experiments typically require significant modification of the land/ocean mask, and therefore require remapping of the river networks across the paleo topography by creating a new runoff directional dataset. This dataset is used by the River Transport Model (RTM) to route river runoff to the ocean.

RTM runs inside the land model, but uses a fixed regular grid that is different from the parent CLM grid (the CLM and CAM models use the same grid, which for CCSM3, is a Gaussian grid, typically at T31 resolution for deep time). The runoff directional forcing file required for RTM is an ASCII file containing latitude, longitude, and an integer value describing the vector (direction) for runoff flow at each RTM grid point. The integer values are numbered from 1 to 8: 1=N, 2=NE (45°), 3=E (90°), etc. See Figure 6 for complete vector directions and integer labels.

The CCSM3 default RTM grid is at 0.5 degree resolution (rdirc.05). However, deep time modelers typically use a 2x2 degree grid. Instructions on how to modify the CLM code to accommodate a 2x2 degree grid can be found in Section 4.3.5.

Figure 6. RTM directional routing key

a. River direction key. The numbers indicate the direction of flow from the center (reference) box.



8↖	1↑	2↗
7←	ref	3→
6↙	5↓	4↘

b. Example of river runoff grid assignments. The numbers correspond to the direction of flow with respect to the grid cell.

3	2	2	1	7	7
1	3	5	7	3	2
5	1	7	8	2	1
6	6	2	8	1	8
7	7	1	2	3	5
8	3	2	1	2	4

c. Directional arrows correspond to the direction of river runoff show in (b). Shaded boxes highlight an infinite loop. See Section 4.3.4.2.

→	↗	↗	↑	←	←
↑	→	↓	←	→	↗
↓	↑	←	↖	↗	↑
↙	↙	↗	↖	↑	↖
←	←	↑	↗	→	↓
↖	→	↗	↑	↗	↘

4.3.4.1 Creating RTM forcing file

- Tool: `rdirc.csh/topo2rdirc.F90`:

The script `rdirc.csh` computes the runoff direction at each land grid point based on the user-provided `topobathy.nc` file containing topography and ocean bathymetry. An ASCII file of runoff vectors is created which is used as input to the RTM at runtime. This tool uses topography to compute the direction of runoff flow.

HINT: The output filename from `rdirc.csh` for your RTM forcing file is simply `fort.10` (Fortran output file), so you may wish to rename your RTM forcing file to something more descriptive. For example, the default RTM forcing file for present day is called `rdirc.05`. An example filename for a paleo run could be: `rdirc_myrun.resolution.date`.

4.3.4.2 Correcting RTM forcing file

- Tools: `check.csh/check_inf_loop.F90`



If your surface topography has any internal basins or large flat regions, infinite loops will result, and `rdirc.csh` will produce another ASCII file with these loops (`fort.11`). An infinite loop is a region from which runoff will never flow out to the coastline, but circulate back to the starting point. If infinite loops are not removed, global freshwater will not be conserved, and undesirable trends in global volume averaged ocean salinity may result. Using a plot of runoff vectors and the list of infinite loops, you must hand edit the runoff vector file to remove all infinite loops. A tool to check for your infinite loops is called `check.csh` and is included in `setup_tools`. Feel free to use your own tool to edit the RTM forcing file. The shaded section of Figure 6c shows an example of an infinite loop.

4.3.4.3 Plotting the vectors on a map

- Tools: `plotdirc.csh`/`rdirc.ncl`

An NCL script to plot your RTM forcing file vectors onto a latitude/longitude map is included in the `setup_tools` tar file. An example plot of a RTM vector directional map generated using `plotdirc.csh` is shown in Figure 7.

4.3.4.4 Iterate

Creating the RTM forcing file and checking for infinite loops is an iterative process. You will need to repeat steps 4.3.4.2 and 4.3.4.3 as many times as needed until you have a clean RTM forcing file. You may want to save your original `fort.10` for comparison with later iterations.

Table 10: Summary Table for `rdirc.csh`

Shell script	<code>rdirc.csh</code>
Source code	<code>topo2rdirc.F90</code>
Inputs	<code>topobathy.nc</code>
Output	<code>fort.10</code> (RTM forcing file used in model) <code>fort.11</code> (infinite loop locations)

Table 11: Summary Table for `check.csh`

Shell script	<code>check.csh</code>
Source code	<code>check_inf_loop_sed.F90</code>
Inputs	<code>fort.10</code>



Output	fort.11 (infinite loop locations)
---------------	-----------------------------------

Table 12: Summary Table for plotdirc.csh

Shell script	plotdirc.csh
Source Code	rdirc.ncl
Inputs	fort.10
Output	Postscript file for vector plot



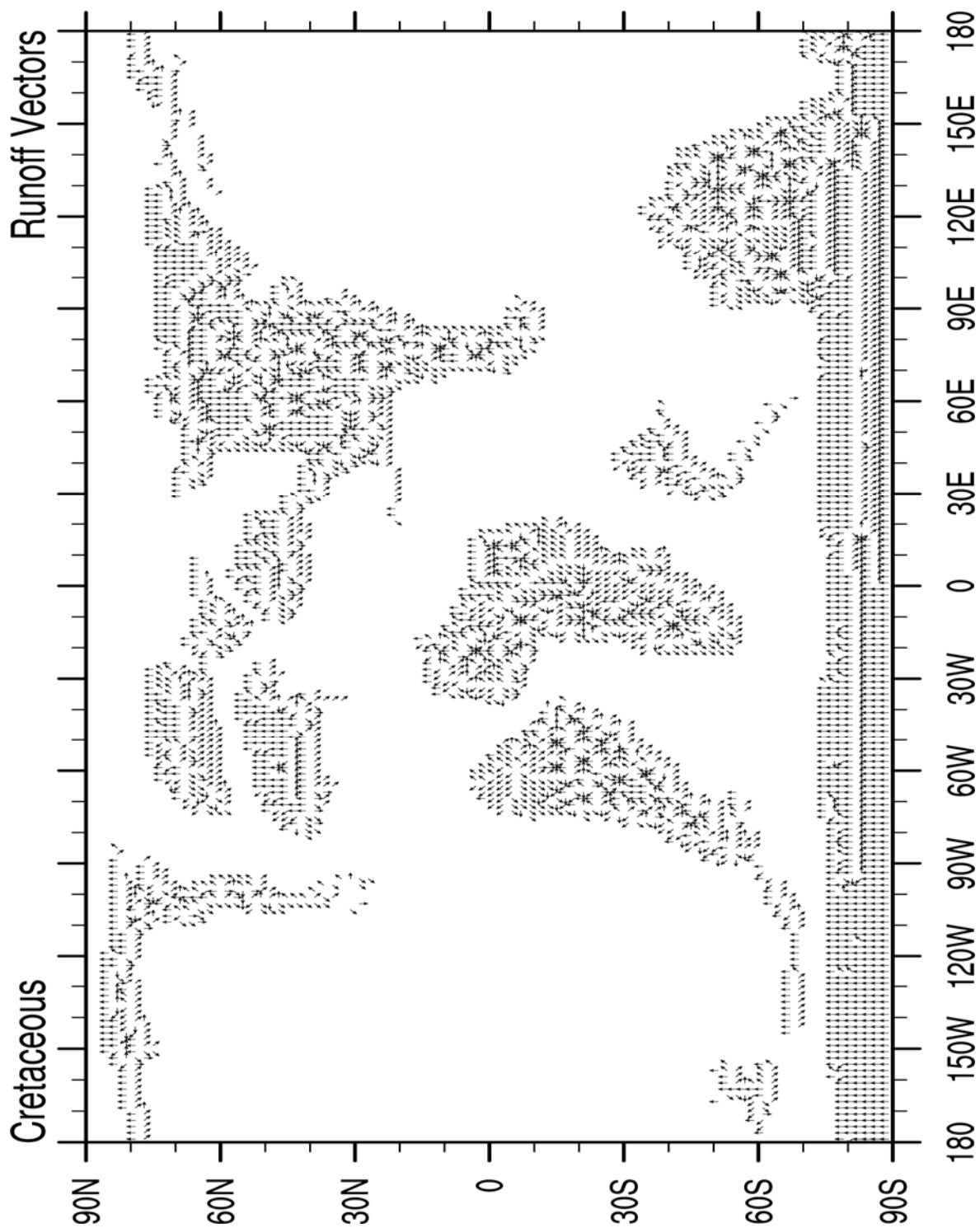


Figure 7. River Runoff

Example plot of runoff vectors for 80 Ma for a 2x2 degree RTM resolution grid.



4.3.5 Other Land Issues

Model dynamics parameters tend to be resolution dependent and can be controlled in the model namelist. These parameters may not need to be altered from the default CCSM3 configuration, but if they do, instructions will be covered in Chapter 5, *Model Setup Scripts and Runtime Issues*.

There are several source code modifications the modeler may need to make.

3. mksrfdatMod.F90

1.1 In the default CLM3 source code, the Ross Ice Shelf is hardwired into the mksrfdatMod.F90 code. If you wish to remove the Ross Ice Shelf, copy mksrfdatMod.F90 to your SourceMods/src.clm directory and search for 'Ross Ice Shelf'. Comment out the entire section referring to setting LAND values on the Ross Ice Shelf to glacier.

1.2 This modification is recommended for all users.

Original:

```
pctwet(i,j) = 100. - pctlak(i,j)
pctgla(i,j) = 0.
```

Modified:

```
pctwet(i,j) = 100. - pctlak(i,j) - pctgla(i,j)
! pctgla(i,j) = 0.
```

1.3 Additionally, if you have lowered sea level and exposed new land along the continental shelf, CLM will automatically define that new land as 'wetland'. If you desire another vegetation designation for these newly created land areas you can modify mksrfdatMod.F90 to fill in these areas (e.g., with nearest neighbor land cover).

4. clm_varpar.F90

If your RTM forcing data set (produced by rdirc.csh) is NOT 0.5 degrees, the modeler will need to edit this code to specify the correct latitudes and longitudes. Search for rtmlon and rtmlat.

All code modifications need to be placed in the \$CASEROOT/SourceMods/src.clm subdirectory of your CCSM3 case. See Appendix 8.4 and CCSM3 user documentation for details on how to create a new case and run CCSM3.

4.4 Ocean and sea ice - Deep Time

Initial and boundary forcing files for the ocean (POP) and sea ice (CSIM) models are listed in the Summary Table (Appendix 8.2). The deep time paleo modeler is responsible for creating the ocean bathymetry, the ocean grid, the ocean region definitions (called the region mask), and the ocean initial conditions. Optionally, the modeler may also specify new locations for diagnostic transport calculations.



Other inputs to the ocean model are of the form of POP's input_templates. These are files read into POP much like a namelist but deliver a wide variety of information.

Forcing files required for the sea ice model (CSIM) are the ocean grid and the ocean bathymetry. A requirement in CCSM3 is that the ocean and sea ice model components share the same grid, which is an irregular POP dipole grid; deep time modelers typically use a nominally 3° ocean/ice grid. Ice initial conditions files are not required for deep time paleoclimate cases. It is recommended that the modeler begin with a 'no ice' state and allow the model to simulate an ice state. The 'no ice' state is set in csim.buildnml_prestage.csh:

```
set no_ice_ic = .false.
```

Creating the ocean boundary forcing (grid, bathymetry, and region mask) is often time consuming and subjective. In this section we give a general overview of the process followed by more detailed steps.

Tips on initial conditions choices and input template changes will be discussed in the detailed-steps sections.

4.4.1 Designing your ocean grid

a. Choose your grid size:

Deep time paleo modelers typically use the low resolution version of CCSM3 (T31_gx3v5; T31 is the land/atmosphere grid, and gx3v5 is ocean/ice grid). We recommend that you choose a CCSM3-supported grid size for your simulation. The most common supported ocean grid sizes are styled after gx3v5 (100 longitudes and 116 latitudes) and gx1v3 (320 longitudes and 384 latitudes). (See the CCSM3 documentation for further details). Although building a new ocean grid with a non-supported grid size is possible, additional changes would need to be made in the ocean and ice source code (Section 4.4.8.2). Examples and tools described in this document are designed for supported ocean grid sizes.

b. Grid pole placement:

The ocean model requires that grid poles be placed over land. Numerically no computation can be done at the convergence point of all longitudes at the grid pole. The ocean model solves this problem by shifting the grid pole away from the geographic pole and placing it over a land mass. (Atmospheric models solve this problem by using numerical filters). Therefore, in CCSM3, if there is no land at the geographic pole, the numerical pole must be shifted over land elsewhere. As long as land exists poleward of ~65°, our tools should be able to create an ocean grid for POP without code modification. However, if no land exists in polar regions, grid creation may be possible but will require extra effort. Contact the deep time paleo liaison for consultation.



Pole placement is a subjective process, however, we offer a few helpful tips.

1. Try to place the grid pole as close to the geographic pole as possible.
2. Try to place the grid pole close to the continental edge without creating spurious land cells around the pole disc.
3. We recommend placing the pole 1-2 land cells from the edge of the continent.

4.4.2 Bathymetry (KMT)

The ocean bathymetry in POP is called the KMT. POP requires the ocean bathymetry (also referred to as ocean topography) to be input into the model as integer values that represent *depth levels* (not ocean *depths*). POP translates the *depth levels* into ocean *depths* using an ASCII file called the 'vertical grid' (e.g., gx3v5_vert_grid). The vertical grid file is resolution dependent. For example, the vertical grid for a gx3v5 size grid specifies 25 vertical depth levels, (KMT=1-25 KMT); whereas the gx1v3 size grid specifies 40 vertical depth levels (KMT=1-40). The vertical grid file has three columns that correspond to (1) ocean layer thickness (cm), (2) midpoint depth (m) of that layer, and (c) the actual depth of the layer (m). Each line of the vertical grid file equates to a KMT level; e.g., in the gx3v5 vertical grid file, KMT=2 equates to a layer thickness of 844 cm, a midpoint depth of 12 m, and an actual layer depth of 16 m (Table 13). KMT=0 denotes land grid cells. We highly recommend using the default vertical grids.

Table 13. Vertical grid file for KMT

Line number= KMT level (line numbers are not included in vertical_grid)	Thickness (cm)	Depth of midpoint (m)	Actual layer depth (m)
1	800.0000	4.0000	8.0000
2	844.0491	12.2202	16.4405
3	929.6631	21.0888	25.7371
4	1053.6499
5			

Note: the first column shows the line number of the vertical_grid file, but is not in the file; i.e., the shaded cells show the actual appearance of the vertical_grid file.

Our tools allow the modeler to create KMT data based on your bathymetry data (contained in topobathy.nc) and place that data onto your model grid. Once the KMT data is placed onto the grid (both horizontal and vertical), the modeler will be required to edit the data to eliminate potential problems for the POP grid and miscellaneous errors generated by the automated process. Modifying the KMT is another subjective process, but we provide some guidance.

- a. Avoid one grid cell wide channels and bays at all levels. The ocean model will not be able to compute flow with one grid box. One typically removes these channels by filling them with land, or widening them to have a minimum width of two grid cell. Note, our tool contains an option to try and remove these channels automatically, but isolated points may still occur.



- b. Channels that are not straight may contain a grid cell width of one where the channel curves (Figure 4). These transition areas will be missed with the automatic checks and must be edited. If not, there will be no flow through the transition area due to zero velocities at both corner (land) cell borders. The recommended change would be to widen the transition area to two ocean grid cells.
- c. Avoid small and shallow bays. Although a two cell wide bay may contain ocean velocities in the middle of the bay, realistically, there may not be enough circulation to fully resolve the ocean flow. Widening these bays may avoid negative salinities (in the case of too much fresh water runoff into the bay) or super-saline bays (in the case of excess evaporation).

4.4.3 Ocean regions

- `Mk_ocninput.csh/modregmsk_edit.f`

The region mask file is a simple binary file that assigns an integer number to each ocean basin. The integer value allows the ocean model to identify the various water bodies where the ocean model is active (lakes are handled in the land model and are considered land cells). For example, the modern oceans such as the Pacific, Atlantic, and Indian Oceans each have a unique region mask number. Enclosed active oceans, such as the Black Sea, are considered marginal seas and are identified as marginal with a negative integer number.

For deep time paleo cases, however, we may not have data to differentiate individual ocean basins, so we simplify the ocean region mask by dividing the domain into two regions: the Northern and Southern Hemispheres.

This code will create your region mask file by assuming your region mask file will contain only the Northern and Southern hemispheres. If this is not the desired mask, you will need to modify the Fortran code to specify your unique ocean regions.

Once regions have been chosen, the modeler will need to modify the ASCII `input_template`, (`gx1v3_region_ids`) to reflect the new regions.



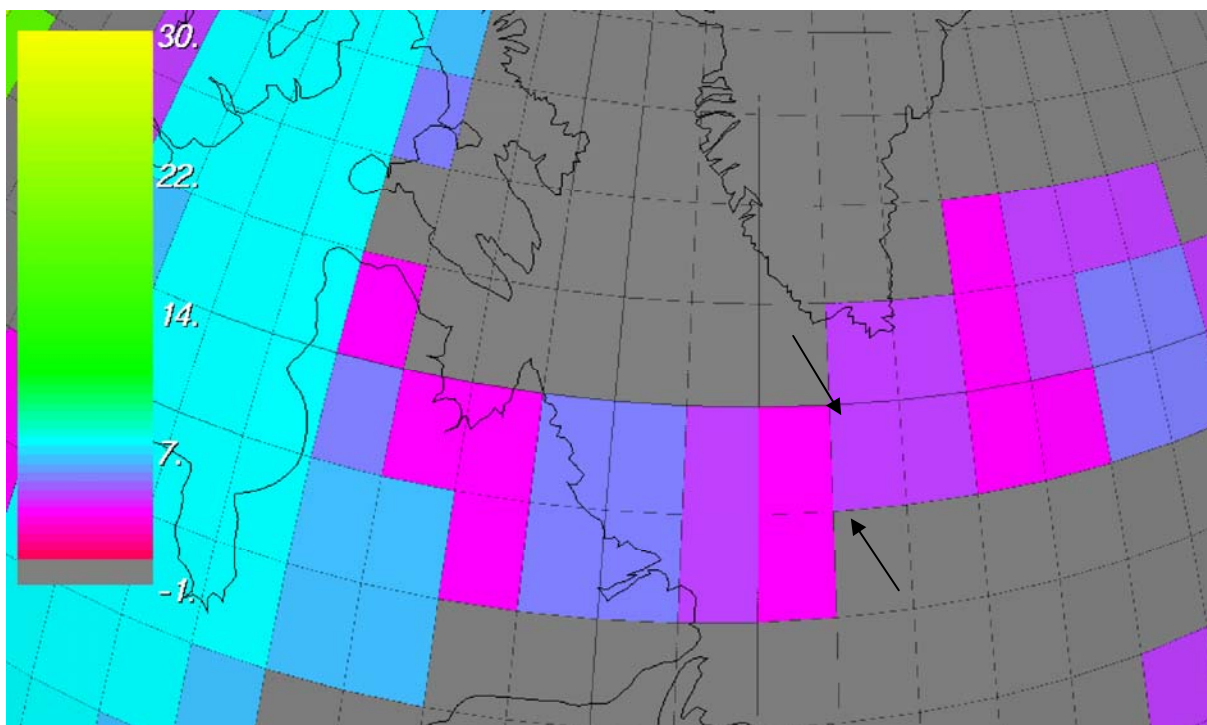


Figure 8. Ocean Grid using kmtEd.

The black arrows show zero velocities at land corners bordering an ocean channel transition area. Each cell is a tracer grid cell with the center of the cell being the tracer grid point. The corners of the cell are the horizontal velocity grid points. (See POP user guide for further grid definitions). Gray cells denote land in this example paleo KMT configuration. Color cells denote active ocean with colors representing integer values of KMT as given by the legend. The black lines show present day continental boundaries relative to the paleo configuration. This image was rendered by a GUI-interface tool called kmtEd. See the tools table and the details section of this chapter for further information on kmtEd.

4.4.4 Ice Initial Condition Details

Ice initial conditions files are not required for deep time paleoclimate cases. We recommend initializing the ice model with a 'no ice' initial state and allowing the model to simulate an ice state. Set 'no ice' in the ice model namelist, called `csm.buildnml_prestage.csh`:

```
set no_ice_ic = .false.
```

4.4.5 Ocean Initial Condition Details



For the ocean model, the modeler has four options for defining initial ocean/ice conditions for startup runs. The choices include, mean, zonal-mean, startup (default), and internal. Details on how to invoke each choice will be discussed in the model build scripts section of Chapter 6.

4.4.5.1 Mean

Initialize with a global, horizontally averaged temperature/salinity depth profile. Initializing with a global volume averaged temperature profile is the recommended method for deep time paleo simulations. Often, very little information on deep ocean temperatures is known, so initializing with a simple depth profile is the easiest method. POP requires this initial file to be a simple ASCII file with a temperature and salinity value for each vertical level. See the example ASCII file (ts_init_guess_gx3_cold.dat) in setup_tools.tar and shown in Figure 5.

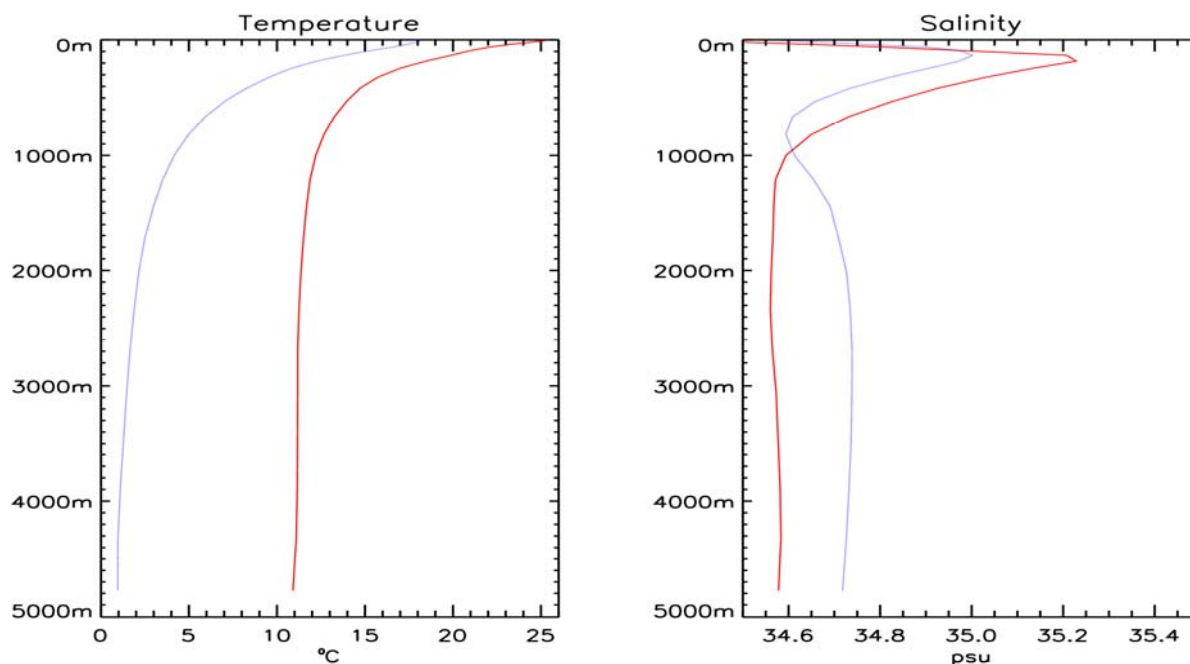


Figure 5. Examples of global horizontally-averaged initial ocean T and S profiles. Blue profile is global horizontally-averaged Levitus T and S; Red profile is horizontally-averaged T and S from a previous CSM1.4 Cretaceous run.

4.4.5.2 zonal-mean

Initialize with a global, zonally averaged temperature/salinity distribution. POP allows a zonally averaged temperature/salinity file to be used for initialization. This file is binary and the format can be found in the CCSM3/POP source code, subroutine initial.F



4.4.5.3 startup (DEFAULT)

Initialize with a full spatially distributed temperature/salinity dataset provided by the user. If the modeler has latitude, longitude, and depth information on temperature and salinity, the ocean model can be initialized in the default configuration with a binary file at startup. This is recommended only if the modeler has this information on an appropriate grid or a grid that is close enough for interpolation. We provide a sample NCL script that will interpolate initial T/S information between two similar grids. This option will be more appropriate for near-modern cases. (See Chapter 3).

4.4.5.4 Internal

Initialize with the default Levitus temperature/salinity profile. Present day temperature and salinity profiles can be computed internally at runtime based on 1992 Levitus data.

4.4.6 Building your ocean grid

Our tools are designed to combine the grid and KMT creation steps. You may modify the provided scripts to suite your programming style. The script `mk_grid.csh` will create your grid using a Fortran program called `ns_dipole.f`, it will create your KMT file using a Fortran program called `paleotop.f90`, and it will convert the binary output files into a netCDF format for easier viewing and modifying. This script may need to be run iteratively depending on how many modifications to your grid are necessary. See the Overview section for guidance on how to create your grid. **Steps 1-3** will describe `mk_grid.csh`.

Once you are happy with your grid, you will then proceed to **Step 4**, modifying your KMT file. Modelers may modify their KMT file with whatever tools proves most useful. We include `kmtEd` and NCL scripts in the `setup_tools`, but some paleo modelers have used tools, such as Matlab, to modify their KMT file. See the Overview section for guidance on your KMT file.

Once you are happy with your KMT file, then proceed to **Step 5** to convert your netCDF KMT file back to the binary file required by POP. We provide a Fortran program called `gridkmt_nc2bin.f90` as well as NCL scripts to accomplish this task.

Finally, proceed to **Steps 6-8** to create your region mask and edit the necessary input templates for your paleo case. All of these steps can be accomplished with `mk_ocninput.csh`.



For all Fortran code, the modeler is responsible for compiling and creating executables. Makefiles for AIX systems are included in the setup_tools.tar file. The scripts are designed to follow a directory structure similar to what is found in preStage.csh (See Appendix). The modeler will need to edit the scripts to point to her/his code, data, and executables.

- **STEP 1: Creating the ocean grid**
- Tool: mk_grid.csh/ns_dipole.f

The script mk_grid.csh sets many variables in the top portion of the script. These variables control settings in ns_dipole.f90, paleotop.f90, and grid_nc2bin.f90 with detailed explanations in **STEPS 1-3**.

Edit mk_grid.csh for the following variables which control ns_dipole.f90 information:

Table 14: mk_grid.csh/ns_dipole.f90 details

Variable	Description	Notes
nx	Number of i grid lines	
nlatn	Number of j grid lines in NH	
nlats	Number of j grid lines in SH	
lonnp	Longitude of grid North Pole	
latnp	Latitude of grid North Pole	
lonsp	Longitude of grid South Pole	
latsp	Latitude of grid South Pole	
dyeq	dy in degrees at Equator ¹	grid box length at equator
dsig	Gaussian e-folding scale at Equator ¹	
jcon	Number of rows of constant dy at poles	
pltgrid	Name of binary plotting grid file ²	Output
popgrid	Name of binary pop grid file ³	Output

¹These variables refer to equatorial enhancement for the POP grid. Default values for the gx3v5 grid are provided in the mk_grid.csh file provided in the setup_tools.tar file.

²The grid plotting grid file is required for paleotopo.f90 and differs from the POP grid file such that it contains one extra latitude grid point, i.e. nlatn+nlons+1

³This file will ultimately be the grid file used in the POP model. The script mk_grid.csh calls this file grid.pop.da, but the CCSM3 setup script for the ocean model calls this file horiz_grid. Variables found in the POP grid file are as follows: ULAT, ULON, HTN, HTE, HUS, HUW, and ANGLE. See the POP user documentation for details.

- **STEP 2: Creating the KMT file**
- Tool: create0.5degree.ncl/mk_grid.csh/paleotopo.f90
- Source: setup_tools.tar



The Fortran program paleotopo.f90 requires the bathymetry data be in 0.5°x0.5° format. In the setup_tools.tar file, we provide an NCL script to interpolate to 0.5° if necessary. Otherwise, the modeler will need to modify the Fortran code.

Run create0.5degree.ncl

Input: topobathy.nc

Output: topobathy_0.5degree.nc

Edit mk_grid.csh for the following information which control paleotopo.f90 information:

Table 15: mk_grid.csh/paleotopo.f90 details

Variable	Description	Notes
nx	Number of i grid lines	
ny	Number of j grid lines	nlats + nlons ¹
nz	Number of vertical grid levels ²	
pltgrid	Name of binary plotting grid file	Input; see ns_dipole
vrgrid	Name of ASCII vertical grid file ²	Input
topo	NetCDF topography data set	Input; e.g., topobathy_05degree.nc
minz	Minimum allowable depth in meters ³	
mink	Minimum allowable KMT value ⁴	
kmtgrid	Binary KMT file ⁵	Output

¹ This value is coded to automatically set to nlats+nlons, you do not need to change if using mk_grid.csh

² The vertical grid is read into POP via the input_templates file. The value of nz and the name of the file will depend on your vertical resolution.

³ The default minimum allowable depth is set to 5 meters. If the ocean depth at any location is equal to or less than minz, then KMT at that location is set to zero. This is one of two criteria used to determine the land/sea mask. The other is that if there is more than 50% land, KMT is set to zero.

⁴ The default minimum allowable KMT value is set to 3 so that there can be vertical exchange between grid boxes. Numerically, mink can be as low as 2, however, we recommend at least 3.

⁵ The binary KMT file to be converted to netCDF, modified/corrected, and ultimately used in the POP model.

This step also produces an elevation file called h.da; this file is NOT used.

- **STEP 3. Convert model grid and KMT to viewable file**
- Tools: mk_grid.csh/grid_bin2nc.f90 or bin2nc_i4_toporegion.090204.ncl

Once the KMT binary file has been created, it can be converted to netCDF for visualization and modification.



- **Choice a: mk_grid.csh/grid_bin2nc.f90**

The mk_grid.csh script calls the Fortran program grid_bin2nc.f90 to create a netCDF file acceptable for the GUI-interface tool kmtEd. This information includes all grid information (i.e. variables in the pop grid file), plus elevation (bathymetry value in meters) as well as the KMT information. Other than the KMT values, all other information in the netCDF file are for viewing purposes only. Only the KMT values will be used in the POP model, the elevation variable is for your diagnostics only. Elevation values are ultimately computed in the model at runtime based on the KMT binary data file.

NOTE: if you notice an error in your grid information, you will need to go back to the ns_dipole step for correction. The binary file generated by ns_dipole is the file used in the model, not the variables in this diagnostic netCDF file.

Edit mk_grid.csh for the following variables which control grid_bin2nc.F90:

Table 16: mk_grid.csh/grid_bin2nc.F90 details

Variable	Description	Notes
nx	Number of i grid lines	
ny	Number of j grid lines	nlats+nlons ¹
cdfgrid	NetCDF grid/KMT file	Output
vrgrid	Name of ASCII vertical grid file ²	Input
kmtgrid	Binary KMT file	Input; see paleotopo.F90
pltgrid	Name of binary plotting grid file	Input; see ns_dipole
popgrid	Name of binary POP grid file	Input; see ns_dipole

¹ This value is coded to automatically set to nlats+nlons, you do not need to change if using mk_grid.csh

² The vertical grid is read into POP via the input_templates file. The value of nz and the name of the file will depend on your vertical resolution.

- **Choice b: bin2nc_i4_toporegion.090204.ncl**

If you do not wish to create a grid/KMT netCDF file with grid_bin2nc.f90, and only want to convert the binary KMT file to a netCDF file, feel free to use the NCL script, bin2nc_i4_toporegion.090204.ncl. This NCL scripts converts both the KMT file as well as the region mask file to netCDF format. (Simply comment out the region mask read/write sections of the code if you do not have a region mask file yet).

You will not be able to use kmtEd if you choose this option.

At the end of the STEP 3, run mk_grid.csh. If you choose STEP3/Choice b, comment out the grid_bin2nc section of mk_grid.csh and run bin2nc_i4_toporegion.090204.ncl.

- **STEP 4: Evaluate/edit KMT**
- Tool: kmtEd/NCL



Refer to the Overview section regarding the Create the bathymetry (KMT) grid for guidance on how to improve your KMT data. There are a variety of tools the modeler can use to achieve this goal.

- **Choice a: kmtEd for low resolution grids**

At NCAR, we typically use kmtEd for the gx3v5 grid size. kmtEd code and sample build/makefile information can be found in the setup_tools.tar file. kmtEd generates a 3D-sphere graphical interface for viewing/editing grid KMT values. This tool is based on eCubed and was developed at Los Alamos National Laboratory (LANL) by John Davis. For further documentation see the LANL website: <http://climate.lanl.gov/Software/ggg/>.

Installing kmtEd requires prior installation of VTK freeware (<http://public.kitware.com/VTK>).

Although there is a fair amount of leg work to install kmtEd, it is an extremely powerful tool to edit the KMT data. The GUI interface is easy to manipulate and the modeler can edit the data with ease. (If you are using kmtEd on the NCAR machines, contact the deep time paleo liaison for location). Once kmtEd is installed, to run, simply type

```
kmtEd -i input_grid.nc -o new_output_grid.nc
```

Enter kmt when queried for your input variable (elevation is default), and simply hit return for the dimension defaults. Consult Section 8.7 for the quick guide on how to operate the GUI. (or see kmtEd/include/controls.h comments) in the code.

Your input_grid.nc is the netCDF file generated by grid_bin2nc.f90. If you do not run grid_bin2nc.f90 and instead opt to use the NCL tool to create the netCDF KMT file, you will not be able to use kmtEd.

- **Choice b: NCL for high resolution grids**

This option is recommended for high resolution grids (such as gx1v3) because kmtEd could be too slow on front-end processing computers.

In the setup_tools.tar file, we provide a sample NCL script to hand edit your KMT file. This technique requires coding the desired KMT changes within the NCL script. It is up to the modeler to determine how best to code the necessary changes. Script change_kmt_example.ncl is provided as an example and a template.

- **Other Choices:**

Ultimately, it is up to the modeler to decide how best to make the necessary modifications. If the modeler has a favorite tool or language, perhaps that will be the best method. Some paleo modelers have used Matlab with a GUI-interface to change the KMT values.

- **STEP 5: Convert from final grid back to binary**
- Tool: gridkmt_nc2bin.F90/NCL



Once you have perfected your new KMT file via the netCDF file, you will need to convert this back to binary for the POP model.

- **Choice a: gridkmt_nc2bin.F90**

Simply compile and run. The program will prompt you for input and output file names.

- **Choice b: NCL**

The example NCL script `change_kmt_example.ncl` writes the new KMT to both netCDF and binary.

- **STEP 6: Create region mask**

- Tool: `mk_ocninput.csh/modregmsk_edit.f`

The script `mk_ocninput.csh` actually accomplishes STEPS 6-9. It creates the region mask based on the `modregmsk_edit.f` code, it creates the region identifiers (`region_id` `input_template`), it creates the diagnostic `input_template` for locales to compute ocean transport, and it grabs all `input_templates` for your grid size and copies them all to a single location with your unique grid name.

You will need to edit `mk_ocninput` to specify your code and data locale, to specify your grid and KMT binary files names, and to specify your new unique grid name. If you have used `gx3v5` for your grid size, (i.e. 100 longitudes and 116 latitudes), it would still be wise to rename your grid to something more appropriate for your case to distinguish it from the default `gx3v5`. For example, `g3vJ` where `g3` would imply the `gx3v5` size, but `vJ` would imply version Jurassic, if your period was the Jurassic.

You will also need to modify the `mk_ocninput.csh` script for each section of the script that deals with STEP 7-9.

In STEP 6, we are modifying the ocean code `modregmsk_edit.f`. Although no changes are necessary to the `mk_ocninput.csh` script itself, edits to the Fortran code are necessary if the modeler requires a region mask other than Northern Hemisphere and Southern Hemisphere. It is up to the modeler's program style to determine how best to accomplish this goal for deep time periods. Each period is unique therefore we cannot provide an all-inclusive algorithm.

- **STEP 7: Create region mask identifiers (input_template)**

- Tool: `mk_ocninput.csh`

Be sure to read the comments in `mk_ocninput.csh`.

Input_templates are ASCII files required at POP's runtime and operate similar to namelist files. Each `input_template` deals with aspects of POP that can be changed by the modeler.

The `region_ids` `input_template` compliments the region mask binary file such that it gives POP information for each ocean region. For each region, the integer value and the ocean basin name are set. If the ocean basin is a marginal sea, the model



requires a location (latitude and longitude and area) to be specified for the redistribution of net freshwater from the marginal seas. Marginal seas are flagged with a negative number. If the ocean basin is not a marginal sea, then these values should be set to 0. Integer values in the region_ids file must be ascending order of the absolute value of the integer. (i.e. 1, 2, -3, 4, etc. where -3 would be a marginal sea). For examples of the present day region ids, go to the POP source code under input_templates.

- **STEP 8: Create diagnostic transport locations**
- Tool: mk_ocninput.csh

Be sure to read the comments in mk_ocninput.csh

The transport_contents input_template is used for diagnostic purposes only. In this file, the modeler can specify ocean locations (straits for example) for ocean transport computations that will be output in the ocean log files.

Grid point i and j and k locations are required. The modeler can also specify whether the section is meridional or zonal and assigns a section name. For examples of the present day transport_contents file, go to the POP source code under input_templates.

- **STEP 9: Rename input templates to your gridname**
- Tool: mk_ocninput.csh

Finally, the mk_ocninput.csh script copies all other input templates from the default location in the POP source code directory and renames them (in addition to your region_ids and transport_contents) according to your unique grid name. These remaining input templates are as follows:

Table 17: mk_ocninput.csh/input templates

Variable	Description	Notes
depth_accel	Information for depth acceleration	No change
scalar_contents	Scalar constant identifiers	No change
history_contents	Snapshot history output variables other than monthly average	Corresponds to history_nml in pop_in
tavg_contents	Monthly average history file contents	Corresponds to tavg_nml in pop_in
movie_contents	Movie history contents	No change. Not in use
vert_grid	Vertical grid	Although renamed, it should be the same as the grid used in mk_grid.csh
pop_in	Main POP namelist	

- See the POP user's guide for further details.
- If the modeler has opted to create a new grid size (i.e. one other than the gx3v5 and gx1v3 nlats and nlons), mk_ocninput_newgridsize.csh should be used. Additional Fortran code will be automatically taken out of the ocean/ice



input_templates source code directories, renamed, and placed with your new input_templates according to the new grid size. See section on **GOTCHAs** (4.4.8) for potential problems. Input_templates should eventually be placed in the SourceMods/src.pop subdirectory for your CCSM3 paleoclimate case. See CCSM3 user documentation for details on how to create a new case and run CCSM3.

- At the end of STEP 9, edit as needed, and run mk_ocninput.csh.

4.4.7 Summary Tables for building the ocean grid

Table 18: Summary Table for mk_grid.csh

Script/Code	mk_grid.csh: ns_dipole.f, paleotopo.f90, grid_bin2nc.f90
Inputs	topobathy_.5degree.nc
Output	grid.pop.da, kmt.da ,h.da ¹ ,gridkmt.nc

¹ h.da is an elevation file output by paleotop.f90. It is NOT used anywhere, so it can be ignored.

Table 19: Summary Table for kmtEd

Script/Code	kmtEd (GUI interface) kmtEd -i input_grid.nc -o new_output_grid.nc
Inputs	gridkmt.nc
Output	edited_gridkmt.nc

Table 20: Summary Table for mk_ocninput.csh

Script/Code	mk_ocninput.csh modregmsk_edit.f, ASCII region_ids, ASCII transport_contents
Inputs	grid.pop.ieee8, kmt.ieeei4
Output	region.ieee4 (binary region mask file) gridname_region_ids gridname_transport_contents gridname_depth_accel gridname_history_contents gridname_movie_contents gridname_scalar_contents gridname_tavg_contents gridname_vert_grid gridname_pop_in

4.4.8 GOTCHAS (Potential Problems)

4.4.8.1 Binary I/O



When reading and writing binary files, you must consider the byte order of the machine you are using to create the binary CCSM input files (e.g., KMT.da etc). For example, NCAR's Linux machines use little-endian format, while NCAR's current supercomputer (IBM AIX) is big-endian. Endianness is the byte ordering used to represent binary data. Binary files written on a big-endian machine will produce garbage when read on a little-endian machine. The binary format will be critical when using SCRIP and `gen_runoffmap`, which require big-endian byte order if running on the NCAR IBM.

The NCL scripts we provide to convert binary files to netCDF files and vice versa, include code to convert binary files from big- to little-endian. Feel free to use these programs as guides.

NetCDF files are machine independent and are therefore unaffected by endianness.

4.4.8.2 Grid Size

If the modeler has constructed a paleo grid with a default grid size (i.e., 100x116), no code modifications will be necessary. If changes to the grid size are desired, be sure to use the `mk_ocninput_newgridsize.csh` script for section 4.4.3, STEPS 6-9. The files `gridname_model_size.F` and `ice_model_size.F.nx.ny.ncat` will need to be placed in the `SourceMods/src.pop` `SourceMods/src.csim` directories, respectively.

GOTCHA: If you are using any of the `OCN_TRACER_MODULES` specified in the `env_run` file (in the CCSM3 case scripts directory), the new ocean `gridname_model_size.F` file will need to be adjusted by hand to account for the additional `OCN_TRACER_MODULES`. Why? The `pop.buildnml_prestage.csh` script automatically makes adjustments to the tracer count applied to `model_size.F` subroutine. If the grid size is not changed, the default action will be to apply the tracer adjustment to the default subroutine. However, if the modeler has changed the `model_size.F` subroutine, the NT count adjustment must be done manually. Failure to do this will result in POP exiting with a tracer count error. See CCSM3 User documentation for explanation of the `env_run` file.



5 Coupler Mapping

CCSM3 is based on a concept that divides the complete climate system into four component models (atmosphere, land, ocean, and sea ice) and a flux coupler. The flux coupler exchanges information with each component model and passes this information along to the other component models.

The CCSM3 component models are built on two primary grids (Table 8.2): (1) the ocean/sea ice grid (usually at gx3v5 resolution for deep time), and the (2) atmosphere/land grid (usually at T31 resolution for deep time). In addition, the river runoff (usually on a 2°x2° grid for deep time) must be routed from the RTM model to the ocean grid. In order to accomplish the transfer of information among component models, fluxes from one model grid must be mapped onto another grid without losing information.

The final step in preparing the model for paleoclimate simulations is to create coupler mapping files. CCSM3 uses these files to exchange information to/from each model component. Because CCSM3 has two distinct component model grids, the modeler must prepare mappings from the ocean grid to the atmosphere grid, and vice versa. Two different styles of mapping files are required; one uses a bilinear interpolation and the other uses a conservative remapping method. A mapping file must also be prepared for the runoff information. The runoff passes directly from the River Routing Model (RTM, within CLM) to the ocean model, POP. RTM runs on a separate grid from CLM, so a separate mapping file is required.

In total, the coupler will use the following mapping files:

- map_atm_grid_to_ocn_grid_aave.nc
- map_atm_grid_to_ocn_grid_bilin.nc
- map_ocn_grid_to_atm_grid_aave.nc
- map_rtm_grid_to_ocn_grid_aave.nc†

† River runoff mapping

To create the coupler mapping files, CCSM3 uses a software package called SCRIP which was developed at LANL. For more information on SCRIP, see <http://climate.acl.lanl.gov/software/SCRIP>.

We provide SCRIP1.4 in our setup_tools.tar file. If the modeler is not using an NCAR machine, SCRIP1.4 will need to be compiled on your local machine. Otherwise, the local NCAR copy/executable can be used and the location can be found in our scripts.

This chapter will discuss how to create the coupler mapping files. We also provide a diagnostic tool to view and check these files.



5.1 Component Model Mapping Files

5.1.1 Two-way atmosphere-ocean grid mapping

- Tool: mk_remap.csh

The script mk_remap.csh creates four component model mapping files:

1. map_atm_grid_to_ocn_grid_aave.nc
2. map_atm_grid_to_ocn_grid_bilin.nc
3. map_ocn_grid_to_atm_grid_aave.nc
4. map_ocn_grid_to_atm_grid_bilin.nc

Of these four, only the first three will be used at runtime in your paleoclimate CCSM3 simulation. Edit the mk_remap.csh script for all name, data and script locations.

Additionally, mk_remap.csh requires the following settings:

Table 21: mk_remap.csh details

Variable	Description	Notes
nx	Number of ocean longitudes	
ny	Number of ocean latitudes	
popgrid	Name of binary ocean grid file	
kmtgrid	Name of binary ocean KMT file	
atmgrid	Name of atmosphere grid ¹	Input
map1_name	Descriptive string for your ocn to atm grid	
map2_name	Descriptive string for your atm to ocn grid	
ocngridname	Descriptive string for your SCRIP ocean domain grid file ²	
ocngridfile	name for your netCDF SCRIP ocean domain grid file ²	Output
map1_filea	name for your netCDF ocean to atm area average file	Output
map2_filea	name for your netCDF atm to ocean area average file	Output
map1_fileb	name for your netCDF ocean to atm bilin file	Output
map2_fileb	name for your netCDF atm to ocean bilin file	Output
map_methodsa	conservative	Do not change
map_methodsb	bilinear	Do not change
normalize_opt	destarea	Do not change

¹ The atmosphere grid files for T31, T42, and T85 are found in the setup_tools.tar file. If the modeler requires a grid other than these grids, code can be found in the SCRIP1.4 source code base. Gaussian grids can be created with convertgauss.f and regular grids can be created with create_latlon.f.

²The netCDF ocean domain grid file is not to be confused with the netCDF grid file created in section 4.4.3/STEP3. The ocean grid file created in mk_remap.csh is written specifically for SCRIP code via the subroutine myconvertPOPT.f90. This file will be used again when remapping to RTM grid to the ocean grid, so you must save this file.



5.1.2 Correct mapping errors

Scrip sometimes generates incorrect mapping weights for grid points near the poles. Usually the incorrect weights (less than zero or greater than 1) occur for a very limited number of grid cells, but because erroneous mapping weights can cause errors in the model, these remapping links should be removed. The standard output diagnostics generated by scrip will list bad weights.

We provide an IDL script, called `correct_map_errors.run`, in the `setup_tools.tar` file to remove these links if they exist. You will need to specify your input mapping file and your new, corrected mapping file in the IDL script. Minimally, this code can be used as a guide and algorithm for the modeler to create a similar script in her/his preferred language/tool.

```
IDL > .run correct_map_errors.run
```

5.1.3 Diagnostics for mapping files: `scrip_test`

Once the component model mapping files have been created, we recommend running a diagnostic test script to transform the mapping files into latitude/longitude map format for easy viewing. The tool `scrip_test` simply utilizes `scrip1.4` code to create this interpolation tool. We provide the driver script `scrip_test.csh` in the `setup_tools.tar` file. The executable file "`scrip_test`" is created when compiling `scrip1.4`. Simply modify the driver script to point to the appropriate directory. If the modeler is local to NCAR, simply point your `scrip_test.csh` to the local `scrip1.4` directory.

Edit `scrip_test.csh` for your mapping file locations as well as diagnostic file names for your output.

5.1.4 Summary Tables for Component Mapping Files

In addition to the summary tables, coupler mapping flowcharts can be found in Section 8.11.

Table 22: Summary Table for `mk_remap.csh`

Script/Code	<code>mk_remap.csh</code> <code>myconvertPOPT.f90</code> <code>scrip1.4</code>
Inputs	<code>grid.da</code> , <code>kmt.da</code> , <code>atmgridfile</code>
Output	<code>ocngridfile</code> <code>map_atm_grid_to_ocn_grid_aave.nc</code> <code>map_atm_grid_to_ocn_grid_bilin.nc</code> <code>map_ocn_grid_to_atm_grid_aave.nc</code> <code>map_ocn_grid_to_atm_grid_bilin.nc</code>



Table 23: Summary Table for correct_map_error.run

Script/Code	correct_map_errors.run/IDL
Inputs (One file at a time)	map_atm_grid_to_ocn_grid_aave.nc or map_atm_grid_to_ocn_grid_bilin.nc or map_ocn_grid_to_atm_grid_aave.nc or map_ocn_grid_to_atm_grid_bilin.nc
Output	map_atm_grid_to_ocn_grid_bilin.COR.nc, etc.

Table 24: Summary Table for scrip_test.csh

Script/Code	scrip_test.csh/scrip1.4
Inputs	map_atm_grid_to_ocn_grid_aave.COR.nc map_atm_grid_to_ocn_grid_bilin.COR.nc map_ocn_grid_to_atm_grid_aave.COR.nc map_ocn_grid_to_atm_grid_bilin.COR.nc
Output	map_atm_grid_to_ocn_grid_aave.diag.nc map_atm_grid_to_ocn_grid_bilin.diag.nc map_ocn_grid_to_atm_grid_aave.diag.nc map_ocn_grid_to_atm_grid_bilin.diag.nc

5.2 Runoff Mapping File

5.2.1 Tool: mk_runoff_remap.csh

Creating the runoff mapping file is a two step process. The first step entails running SCRIP to create a mapping file for RTM grid to ocean grid. The second step requires correcting any errors generated in step one and smoothing the final mapping file. The second step is accomplished with the gen_runoffmap tool.

The script mk_runoff_remap.csh is similar to the other coupler mapping scripts but only creates one mapping file using the conservative remapping method.

Edit the script for name, data, and script locations (Table 25).

Table 25: mk_runoff_remap.csh details

Variable	Description	Notes
ocngridfile	Name for your netCDF SCRIP ocean domain grid file ¹	Input
lndgridfile	Name of RTM grid ²	Input
map1_file	Name for your netCDF RTM to ocn grid ³	output
map1_name	Descriptive string for your RTM to ocean grid	
map_method	Conservative	Do not change
normalize_opt	Destarea	Do not change

¹The netCDF ocean domain grid file is not to be confused with the netCDF grid file created in section 4.4.3/STEP3. The ocean grid file is created in mk_remap.csh and is written specifically for SCRIP code via the subroutine myconvertPOPT.f90. This file is used as input to mk_runoff_remap.csh. See Section 4.5.1.1. for details.



² The RTM grid files for .5x.5, 1x1, and 2x2 are found in the setup_tools.tar file. If the modeler requires a grid other than these grids, subroutine create_latlon.f can be found in the SCRIP1.4 source code directory. RTM requires that the runoff grid have longitudes spanning from -180 to +180 which is consistent with CLM code. Remember that RTM is run within CLM.

³ The output mapping file map_rRTM-resolution_to_ocngrid.nc will be used as input in the second step to creating the runoff mapping, i.e. running gen_runoffmap. This will be discussed in section 4.5.2.2.

5.2.2 Tool: gen_runoffmap

The tool gen_runoffmap is a Fortran code that corrects errors generated by mk_runoff_remap.csh/SCRIP that erroneously place runoff values over land instead of ocean. All runoff values need to be mapped to ocean points. In addition to relocating runoff over land to the ocean, gen_runoffmap smoothes the final RTM to ocean mapping file such that runoff is evenly distributed around the coastlines. If runoff values over ocean are not smoothed enough, ocean sea surface salinity values may become too fresh at these grid points. For further details, see the README file in the setup_tools.tar file under the gen_runoffmap directories, (highly recommended).

To run gen_runoffmap, the modeler will need to compile the code and prepare the namelist and build/run scripts. We provide examples that work on the NCAR IBM supercomputer. Modifications may need to be made if running elsewhere.

- **STEP 1. Edit the namelist (Table 26)**
- **STEP 2: Build and run gen_runoffmap.**

The build and run scripts are found in the setup_tools.tar file under the gen_runoffmap directories. Edit these scripts for data, code, and script locations and run the code.

- **STEP 3: Check the final RTM_to_ocean runoff mapping file.**

NCL and IDL tools are provided in the setup_tools.tar file in the gen_runoffmap/tools subdirectory to test the runoff mapping file. If the modeler created a new RTM grid, the most common error is using a grid spanning from 0 to 360, rather than -180 to +180. If this is the case, your runoff map will be transposed 180 degrees. The mapping file can be tested before running the model by using the tools provided. The NCL script creates a binary file containing the ocean grid area (T grid, called TAREA). The IDL script will take a user-provided point source of runoff (i.e., i and j location) and map that runoff from the RTM grid to the ocean grid. The output is a netCDF file and can be used to identify whether or not the runoff is going to the correct location on the ocean grid.



This information can also be gleaned from the coupler history files at runtime. Be sure to specify a coupler history file frequency in the env_run CCSM3 file. More information on coupler history files will be discussed in Section 7.

Table 26: gen_runoffmap namelist

Variable	Description	Notes
file_orig	Name of your initial runoff mapping file ¹	Input
file_corr	First pass mapping filename	Do not change
file_unsorted	Second pass mapping filename	Do not change
file_new	Name of your final mapping file ²	Output
title	Descriptive string for your final runoff mapping file	
file_sources	Unset	Do not change
eFold	e-folding distance (m) to apply to your final runoff mapping file ³	
rMax	Radius of influence (m) for smoothing ³	

¹This file is generated with mk_runoff_remap.csh

²This is your final RTM to ocean grid mapping file and is to be used in the model at runtime.

³ Efold and rMax will control how much smoothing occurs along the coastlines. The larger values will produce more smoothing. Default values are eFold = 1000 km, and rMax = 500km. Note also that more smoothing will require more computation time.

Table 27: Summary Table for mk_runoff_remap.csh

Script/Code	mk_runoff_remap.csh/scrip1.4
Inputs	ocngridfile lndgridfile
Output	map1_file

Table 28: Summary Table for gen_runoffmap

Script/Code	runoff.ibm.run/gen_runoffmap
Inputs	runoff_map.nml (namelist) file_orig (map1_file)
Output	file_new (example output name = map_r2x2_gx3Paleo_<date>.nc)

5.3 Orbital forcing

Orbital parameters are specified in the coupler namelist at runtime. The modeler must determine orbital parameters appropriate for the time period of interest. Orbital parameters can be modified in two ways depending on whether you are modeling a time period older or younger than 1 million years (Ma).

- a. Time periods < 1 Ma, use 'orb_year'



Orbital year is expressed as 1950-[time_period_of_interest].
For example, for 506ka, orb_year = -504050

b. Time periods > 1Ma, use the following three parameters

Table 29: orbital parameters for > 1Ma

Variable	Description
orb_eccen	Eccentricity
orb_obliq	Obliquity
orb_mvlep	Moving vernal equinox

For pre-Quaternary time periods, eccentricity, obliquity, and the moving vernal equinox must be expressed individually. Details on how these variables are computed can be found by reviewing the code in the CCSM3 csm_share subdirectory.



6 CCSM3 Component Model Scripts and Runtime Issues

This chapter will cover script and runtime issues specific to each of the component models. It is the modelers responsibility to learn how to create new default cases for the CCSM3 using the tools `create_newcase` and `configure`. See the CCSM3 user documentation for details, or see Appendix Section 8.4 for an overview.

For paleoclimate modelers, `create_newcase` must be used with one of the default resolution options. Supported resolutions can be found by running `create_newcase` with the `-help` option. Deep time paleo modelers will typically use the `T31_gx3v5` atmosphere/ocean configuration. Near-modern modelers may use `T42_gx1v3` or `T85_gx1v3`. We have recommended in this document that modelers design their experiments with the supported grid/resolution sizes, (i.e. T31 for CAM/CLM, and 100x116 for POP/CSIM). However, as stated in Chapter 4, the modeler may change the name of the ocean grid to reflect the unique grid characteristics (such as pole location and KMT position) even though the resolution may be the same as the default grid (i.e. 100x116).

For example, if you have designed your CCSM3 grids to match that of the default resolutions, `T31_gx3v5` (T31 CAM/CLM and 100x116 POP/CSIM), but have called your ocean grid a unique paleo name, i.e., `gx3Paleo`, the modeler will simply need run `create_newcase` using the `T31_gx3v5` resolution tag, and then edit/adjust the namelist (`[].buildnml_prestage.csh`) scripts appropriately. If the modeler has chosen to change the default grid resolution for the ocean, (for example, 100x120), the modeler will need to run `create_newcase` with the `T31_gx3v5` resolution tag, but also apply the necessary modifications to the ocean/ice model (see Chapter 4) and edit the namelists appropriately.

6.1 Script and Forcing Locale Philosophy

Ultimately, it is up to the modeler to decide how to handle script changes and where the forcing/initial files reside. This may be dependent on local filesystems and disk space. At NCAR, we typically place the scripts (`Buildnml`, `SourceMods`, etc.) on our home directory and place the forcing and initial files on a data space where our space quotas are higher. Regardless of where you decide to place your files, the paleo modeler will need to modify the `Buildnml_Prestage` setup scripts for all component models to point to the appropriate forcing and initial file locations.



6.2 Atmosphere Script and Runtime Issues

6.2.1 CAM3 Datasets

The modeler will need to point to the appropriate forcing datasets for the time period of interest. Typical forcing datasets changed for paleoclimate modelers include but aren't limited to:

Table 30: Cam forcing datasets

Variable	Description	Notes
bndtvo	ozone dataset	May be changed for Near-modern
bndtvaer	aerosol mass ratios	May be changed for Near-modern
datinit	initial condition file	

6.2.2 CAM3 Namelist Parameters

The modeler will need to adjust the appropriate trace gases, solar constant, and aerosol optical depth properties in the namelist section of the setup script. See Chapter 4 for details on the physical forcing namelist parameters.

Other common changes to the namelist include history file modifications (fincl) and time step adjustments (dtime).

Dynamical instabilities (CFL violations) can often be cured by lowering the time step. NOTE that the CAM time step must equal the CLM time step. Also, because dtime is stored in the CAM and CLM restart file, if the time step does need to be changed, a new hybrid run must be created. (A branch or restart run will not work).

For further details on namelist options and CCSM3 runtypes, see the CAM and CCSM3 User documentation, respectively.

The namelist is created within the cam.buildnml_prestage.csh script and is called atm.stdin.



6.3 CLM3 Script and Runtime Issues

6.3.1 CLM3 Datasets

The modeler will need to point to the appropriate forcing datasets for the time period of interest by changing `clm.buildnml_prestage.csh`. Please read Chapter 4 for the discussion of when it is appropriate to use the raw `mksrf` files modified by the tool `convert_mksrf.f90`, or created by `paleo_mkraw`, and when is appropriate to use the default `surface_data` forcing file.

Typical forcing datasets used for CLM are specified in Table 31 for the default scripts. These names may need to be replaced with the appropriate name and/or location specific to your paleo needs. Be sure to look for all instances in the build script `clm.buildnml_prestage.csh`.

Table 31: CLM3 forcing datasets

Namelist variable	filename	Type	Description/filename
<code>fsurdat</code> ¹		netCDF	Surface data file
<code>finidat</code> ²		netCDF	CLM3 initial condition file
<code>fpftcon</code>	<code>pft-physiology</code>	ASCII	PFT physiology specifications
<code>frivinp_rtm</code> ⁴	<code>rdirc.05</code>	ASCII	River transport model initial file
<code>mksrf_fsoicol</code> ⁵	<code>mksrf_soicol_clm2.nc</code>	netCDF	soil color
<code>mksrf_flanwat</code>	<code>mksrf_lanwat.nc</code>	netCDF	land water (lakes, wetlands)
<code>mksrf_fglacier</code>	<code>mksrf_glacier.nc</code>	netCDF	glacier
<code>mksrf_furban</code>	<code>mksrf_urban.nc</code>	netCDF	urban
<code>mksrf_flai</code>	<code>mksrf_lai.nc</code>	netCDF	leaf area index (LAI)
<code>mksrf_fvegtyp</code>	<code>mksrf_pft.nc</code>	netCDF	plant function types (PFTs)
<code>mksrf_fsoitex</code>	<code>mksrf_soitex.10level.nc</code>	netCDF	soil texture

¹The surface-data set is created at runtime using the `mksrf` datasets if `fsurdat = ''`. This is typically done for deep time cases and only needs to be done upon the first model submission. Once the surface-data file is created and corrected, `fsurdat = 'surface-data.atmres_ocnres.date.nc'`. See section 5.3.3 for step by step instructions.

²If the modeler is using a CLM initial condition file using spun-up CLM state variables, this script variable is set. Otherwise, set `finitdat = ''`, for arbitrary initialization.

³ Unless the `pft-physiology` file was modified, the default file/location may be used. Otherwise, specify the unique filename and location.

⁴The default name for the RTM initial condition file is `rdirc.05`. Replace this string with the unique string name for your paleo case.



⁵The 'mksrf' datasets are considered raw land use information. The configure script automatically places the present day raw datasets into the CLM script and namelist. If the paleo modeler is using mksrf raw datasets generated from paleo_mkraw, or modified present day mksrf datasets, these new unique data sets need to be specified in the script/namelist instead of the default files. The presence of these files in the namelist DOES NOT guarantee their use. These files are only used if fsurdat= ''. See 6.3.3 for further information.

6.3.2 CLM3 Namelist Parameters

Dataset names in Table 31 are specified in the namelist. Change appropriately.

Other common changes to the CLM3 namelist include history file modifications (hist_fincl) and time step adjustments (dtime). If the dynamical time step is changed, the RTM time step may also need to be adjusted (rtm_nsteps). RTM is typically called every 3 hours, so if dtime is modified, rtm_nsteps must also be modified to preserve the 3 hour call frequency.

NOTE: The CLM time step must equal the CAM time step. See the atmosphere script and runtimes section 6.2.2.

For further details on namelist options, see the CLM3 User documentation.

The namelist is created within the clm.buildnml_prestage.csh script and is called lnd.stdin.

6.3.3 More details on 'mksrf' versus surface-data

The following set of steps review a typical sequence for the paleoclimate modeler who has created all new mksrf raw datasets.

STEP 1: Edit clm.buildnml_prestage.csh

- a. fsurdat = ' '
- b. mksrf files specified as noted in section 6.3.1

STEP 2: After modifying all scripts as appropriate, run CCSM (startup) for 5 days.

STEP 3: Find the newly created surface-data_clm_resolution.nc and copy it to a working location. Correct/modify as required, (i.e. remove erroneous wetlands if necessary).

STEP 4: Copy final version of the surface-data set to your forcing file location and rename the file to a unique name descriptive of your time period and coupled resolution.



For example, surface-data_atmgrid_ocngrid_timeperiod_date.nc (e.g., surface-data_64x128_gx1v3_21ka_100122.nc)

STEP 5: Point to the new surface dataset in your clm.buildnml_prestage.csh script.

Namelist	Filename
fsurdat	surface-data_atmgrid_ocngrid_timeperiod_date.nc
mksrf_glacier.nc	mksrf_glacier_timeperiod.nc
mksrf_lanwat.nc	mksrf_lanwat_timeperiod.nc
mksrf_pft.nc	mksrf_pft_timeperiod.nc
....

STEP 6: Restart your CCSM job submission with your clm.buildnml_prestage pointing to your final surface-dataset.

Note: For near-modern cases, the modeler may choose to modify the default CLM3 surface-dataset rather than create new mksrf datasets. In this case, STEPS 4-6 can be applied to the modified surface-dataset. This method will only work if the modeler has NOT modified the model grids or KMT.

Note: The modeler must modify the default CCSM3/CLM3 surface dataset appropriate for their desired COUPLED resolution. Stand-alone CAM/CLM surface-datasets will not work in the coupled model, nor will surface-datasets appropriate for versions of CCSM other than CCSM3.

6.4 Ocean Script and Runtime Issues

The pop.buildnml_prestage.csh script looks a bit different than the other component model scripts. Be sure to go over all the c-shell programming and modify location pointers and file names for your forcing and initial files.

6.4.1 POP Forcing Files

Go to the section of the script that acquires initial/boundary datasets and edit the following according to your file names and locations.

Table 32: Ocn forcing datasets changes for paleo

Variable	Description	Example name	Format
horiz_grid	POP grid	myrun.horiz_grid.ieeer8	binary, double
region_mask	Ocean regions (e.g., Pacific, North Atlantic)	region_mask_myrun.ieeei4	binary, integer
bathymetry	KMT file	myrun_kmt.ieeei4	binary, integer

Read all other forcing files from the default location.

6.4.2 POP Initial condition file



The initial condition file is specified with the string INIT_TS_FILE.

The initial condition option is specified with the string INIT_TS_OPTION. The INIT_TS_OPTION specification is handled differently depending on which option is applied. See Chapter 4 for details on options.

Option	Setting
mean	set INIT_TS_OPTION = mean ¹
zonal-mean	set INIT_TS_OPTION = zonal_mean ¹
startup	don't change, use default INIT_TS_OPTION = \$runtype ²
internal	set INIT_TS_OPTION = zonal-mean ¹

¹Options 1,2, and 4 will require changes to the commands.sed section of the pop.buildnml_prestage.csh script where INIT_TS_OPTION is set. For ease of coding, we recommend adding the following lines to your script after runtype has been defined:

```
set ictype = $runtype
if ($runtype == startup) then
set ictype = <insert option type>
endif
```

Next, within the command.sed section, change the definition of INIT_TS_OPTION to the following line:

```
S#INIT_TS_OPTION#$ictype#
```

²The default pop.buildnml_prestage.csh script for a startup run assigns INIT_TS_OPTION = \$runtype. For a startup case, \$runtype = startup. Remember that the only run requiring an initial condition file is a startup run.

6.4.3 POP Input templates and pop_in

The default input_templates specification will use the default grid resolution. For example, for a gx3v5 resolution simulation, the region_ids file will be listed in the pop.buildnml_prestage.csh script as gx3v5_region_ids. Edit the script to point to your unique input_templates filenames. These files should be placed in the SourceMods/src.pop directory.

The pop namelist, pop_in, is treated the same way as the input_templates in the pop setup script. Modify accordingly.

Be careful in the script to only change the pointers to \$my_path (i.e. the SourceMods/src.pop directory) for the input_templates and pop_in only.

The pop_in file controls a wide variety of POP issues including various parameterization choices, history write frequencies, and MOC (meridional overturning circulation) diagnostics. We highly recommend the paleoclimate modeler review the various options and chose the parameterizations most appropriate for the science questions being posed by the experiment.



NOTE: The default choice for the "sw_absorption" in pop_in is "chlorophyll". The chlorophyll dataset is designed for present day geography, therefore, for deep time paleo cases, the choice "jerlov" is more appropriate. See the POP user guide for more details on the pop_in file.

6.4.4 Other Issues

As with all the model components, the results need to be carefully checked for potential problems.

6.4.4.1 Specifying MOC in pop_in

The default gx3v5_pop_in will automatically configure your pop2 namelist to compute the meridional overturning circulation (MOC) *with the assumption of modern day geography*. This will likely cause the model to fail. To turn this option off, change the following parameters in the transports_nml section of the pop_in file.

```
&transports_nml
  moc           = .false.
  n_heat_trans  = .false.
  n_salt_trans  = .false.
```

To automatically compute MOC on the desired paleo grid, modify the transports_nml again to define the user-specified auxiliary zonal (regular) grid unique to your run. In the transports_nml section of the pop_in file, set the following parameters:

```
&transports_nml
  lat_aux_grid_type      = 'user'
  lat_aux_begin          = [insert beginning latitude of regular grid]
  lat_aux_end            = [insert ending latitude of regular grid]
  n_lat_aux_grid         = [insert number of latitude points for regular grid]
  n_transport_reg        = 1 (compute global eularian MOC only)
```

For a generic, low resolution regular grid, typical values for lat_aux_begin, lat_aux_end, and n_lat_aux_grid are -90, 90, and 90, respectively. Parameters "moc", "n_heat_tran", and "s_heat_trans" remain ".true.".

After applying the above specifications to paleo pop_in, the model code will only compute the eularian component to the global MOC. If computation across all paleo ocean basins (as defined by your region mask file) is desired, calculating MOC for each basin is best completed using offline code. Significant POP code modification would be required to do this as a model diagnostic and is not recommended.

For more detail on how to change the pop_in file for MOC, see the CCSM3 POP user's guide, section 4.2.3.

6.4.4.2 Problems arising from KMT and grid errors

- Tool: cmpRegionMask2KMT.ncl
- Source: setup_tools.tar



As mentioned in Chapter 4, take care to create your KMT and grid following the guidelines discussed. Extremely narrow, shallow inlets and straits will produce unphysical salinities.

Another common problem for near-modern cases is a mismatch between the KMT mask (land/ocean grid points) versus the region_mask. If the modeler has modified a present day region_mask file, rather than creating a new one from scratch, the model WILL NOT RUN if these masks do not match exactly. The tool cmpRegionMask2KMT.ncl will compare the region_mask and the KMT for compatibility.

6.4.4.3 Problems arising from runoff mapping errors

Sometimes problems with the mapping files will be evident initially in the ocean model. The most common problem lies with errors in the runoff to ocean mapping. (Other mapping issues will be discussed in the coupler section).

Review the output diagnostics log files and check that global average salinity is conserved. There will be some initial change due to building of sea ice and the draining of rivers into the ocean but over the long term, there should be little drift. Another way to view salinity trends is to make plots of the global volume averaged salinity. (NCL scripts provided in the setup_tool.tar file). If a trend stronger than .002ppt/decade is observed, then the runoff mapping is not conservative or may be incorrect.

6.4.4.4 POP Model Instabilities

If the model crashes in POP with a CFL violation, a common solution is to lower the time step. Unlike the atmosphere and land models, this can be accomplished without creating a new hybrid case and can be simply done with a restart run.

Edit the pop.buildnml_prestage.csh script and increase the variable DT_COUNT found in the commands.sed section. DT_COUNT is the number of time steps within a 24 hour period. For gx3v5 resolutions, the default DT_COUNT = 12 (every 2 hours). For gx1v3 resolution, the default DT_COUNT = 23. (~ 1 hour). Simply increase the DT_COUNT to accommodate your needs. (Common DT_COUNTS are 12, 23, and 40). TIME_MIX_FREQ does not need to be changed.

6.5 Ice Script and Runtime Issues

6.5.1 CSIM Datasets



The ice model requires the ocean grid and KMT file, although the `csim.buildnml_prestage.csh` refers to these files with a slightly different name. Edit this script to point to the appropriate names and locations.

File	Description
<code>data.domain.grid</code>	pop grid file (i.e. <code>grid.pop.da</code> , binary)
<code>data.domain.kmt</code>	KMT file (i.e. <code>kmt.da</code> , binary)

6.5.2 CSIM Initialization

Unless the modeler has a CSIM restart file appropriate for the model grid and KMT, we recommend initializing the model with a 'no ice' state. The model will grow or melt ice as the climate defines and will establish equilibrium during the model integration.

To initialize with a 'no ice' state, set the variable `no_ice_ic` to true in the `csim` namelist (`csim.buildnml_prestage.csh`).

```
set no_ice_ic = .true.
```

6.5.3 CSIM Namelist Parameters

The ice model namelist rarely needs to be adjusted. However, if there is a lot of ice being produced and the ice model crashes with an instability, the dynamical time step in the namelist can be modified, `ndyn_dt`.

The default value for `ndyn_dt` is 1. This signals CSIM to compute one dynamical time step for every thermodynamic time step. By increasing this number to two, the modeler is signaling the model to compute two dynamical time steps per thermodynamic time step.

The namelist is created within the `csim.buildnml_prestage.csh` script and is called `ice_in`.

6.6 Coupler Script and Runtime Issues

6.6.1 CPL6 Datasets

The coupler script, `cpl.buildnml_prestage.csh`, requires four coupler mapping files. Edit the script for name and locations.

Filename	Description
<code>map_a2of_file</code>	atmosphere to ocean conservative mapping (fluxes)
<code>map_a2os_file</code>	atmosphere to ocean bilinear mapping (state variables)
<code>map_o2af_file</code>	ocean to atmosphere conservative mapping (fluxes+state)
<code>map_r2o_file</code>	RTM to ocean conservative mapping



6.6.2 CPL6 Namelist Parameters

Orbital parameters are set in the coupler namelist. Details are in section 4.5.3. The namelist is created within the `cpl.buildnml_prestage.csh` script and is called `cpl.nml`.

6.6.3 Other Issues

Problems with the mapping files can be spotted by reviewing the coupler history data. If the atmosphere to ocean (and vice versa) mapping is incorrect, clues can be found in the various variables written to both the instantaneous and averaged coupler history files.

Coupler mapping at the poles is one area that can be problematic. The coupler computes the ocean grid areas with SCRIP with a slightly different method compared to the method used in the POP code directly. The ratio between these two areas is found in the coupler instantaneous (hi) history file and is called `areafact_o_cpl2comp`. (There are area ratios for all model components, but it is typically the ocean/ice ratios that cause problems). These ratios should be close to 1 and usually range from .98 to 1.02. A range with an error of greater than 10% should be evaluated as a potential problem. When the differences in the grid areas are large, information will be lost and errors will accumulate when mapping from one grid to another via the coupler.

To produce coupler history files, edit the **env_run** file for the following variables:

Variable	Options
HIST_OPTION	monthly, nmonths, daily, ndays, or nstep
AVGHIST_OPTION	monthly, nmonths, daily, ndays, or nstep
HIST_N	frequency of history file creation
AVGHIST_N	frequency of history file creation

See the CCSM3/CPL6 documentation or the `env.readme` file for further details.



7 Diagnostics

7.1 Log files

One of the best tools to evaluate the progress of the model is to review the component model log files, which can be found in the model run directory under the component model subdirectories. The log file naming convention is: model.log.date-timestamp. The individual log files contain the standard output for each model component. It is in these log files that error messages, prints, and various other diagnostics are found.

7.2 Diagnostic flags in env_run

If the model is crashing and the modeler wishes to receive more diagnostic output, edit the env_run to increase the level of output diagnostics found in the coupler. Variables to use include INFO_DEBUG, DIAG_OPTION, and DIAG_N. See the env.readme file for details.

7.3 A few words about post-processing

Processing and analyzing the CCSM3 model output can be a daunting task.

One strategy for consolidating model output is to use your preferred tool (i.e., programming language, interpretative language, or shell scripting) to post-process the output into annual averages, annually-averaged timeseries, climatological means, etc.

Links to CCSM3 post-processing diagnostic packages for all four model components are available upon request, but NCAR offers no user support for these post-processing tools.



8 Appendix

8.1 Setup Tools

All tools are contained in the setup_tools.tar file. Please contact the paleo liaison to download this file.

To untar the setup tools on your local machine:

```
mydir: tar -xvf setup_tools.tar
```

For the preStage.csh tool, see the preStage subdirectory in this distribution. The script preStage.csh is commented and self-explanatory.

8.2 Summary Table for Initial Conditions and Forcing Files

Table 33: Summary Table for Initial Conditions and Forcing Files

	T31, T42, T85 (or FV)		Cpl	gx1v3, gx3v5	
	Atm	Lnd + RTM ¹		Ocn	Ice
Forcing files	aerosol optics aerosol mass absorption/emiss ozone	pft_physiology rdirc.<res> mksrf_<>.nc or surface_dataset.nc ²	a2o aave ³ o2a aave a2o bilin map_rdir_ocr.nc ⁴	kmt(bathymetry) grid.da pop_in ⁵ region_ids transport_contents vert_grid depth_accel	kmt grid.da
Initial files	cam1	clmi or arbitrary initialization		temperature salinity	no ice or modern or create own ice
Physical forcing (namelist)	Solar constant Trace gases Aerosol scaling		orb_yr or orb_eccen orb_mvlp orb_obliq		
Coupler mapping ⁶	Atm/Lnd grid>Ocn/Ice	RTM grid->Ocn/Ice grid	Scrip mapping ³	Ocn/Ice grid- >Atm/Lnd grid	Ocn/Ice grid-> Atm/Lnd grid

¹RTM is on a fixed regular grid (i.e., .5, 1, or 2 degree) with longitudes ranging from -180 to +180

²Surface datasets are created at runtime in CCSM3.0. Run the model for 1 day; remove wetlands as needed (where there is a mismatch between the landmask and ocean grid)

³Mapping tool (scrip) creates weights between grids:



aave: conservative mapping; used for fluxes (choppy fields)

bilin: smoothed; used for state variables (e.g., temperature)

⁴Runoff to ocean smoothing can be tricky

⁵pop_in is the namelist for the ocean model. This file contains various options for the ocean physical parameterization. Detail can be found in the POP users guide.

⁶See the CCSM3 CPL6 documentation for more details

(<http://www.ccsm.ucar.edu/models/ccsm3.0/cpl6/>)

8.3 Useful links

For general information on the CCSM3 model see *The Journal of Climate Special Issue, 1 June 2006*. More information relevant to model setup issues can be found in the following documents:

CCSM3.0.1 General Documentation Page: <http://www.ccsm.ucar.edu/models/ccsm3.0/>

CCSM3 FAQ Page: http://www.ccsm.ucar.edu/models/ccsm3.0/ccsm3.0_faq.html

8.3.1 Component Model User Guides:

CAM3: <http://www.ccsm.ucar.edu/models/atm-cam/>

CLM3: <http://www.ccsm.ucar.edu/models/ccsm3.0/clm3/>

CSIM5: <http://www.ccsm.ucar.edu/models/ccsm3.0/csim/>

POP1.4.3: <http://www.ccsm.ucar.edu/models/ccsm3.0/pop/>

CPL6: <http://www.ccsm.ucar.edu/models/ccsm3.0/cpl6/>

8.3.2 Web sites useful for processing and viewing data files:

CCSM Support Page (NCL help): <http://www.ccsm.ucar.edu/support/>

NCO User Guide and source can be found at: <http://nco.sourceforge.net/>

Ncview: http://meteora.ucsd.edu/~pierce/ncview_home_page.html



8.4 QuickGuide: CCSM3.0 Setup

See CCSM3 model documentation for further details,
<http://www.cesm.ucar.edu/models/ccsm3.0/#docs>

Examples are for NCAR's bluefire machine:

1. Go to CCSM3 source code directory.
 cd to the scripts directory to create the automated runscripts for your simulation:
 - o create_newcase -help (for options)
 - o create_newcase -case /mypath/b30.myrun -mach bluefire -res T31_gx3v5 -compset B
 where:
 - res = model resolution
 - mach = machine name
 - compset = what components are active
 - b30.myrun = CASEID
2. Edit env_conf (set: startup, hybrid, branch run)
 - Branch: uses exact restart files of previous run.
 - Hybrid: uses restart and initial files.
 - If you are running a hybrid or branch run:
 - Branch runs: pre-stage the restart files in your /ptmp/USER/archive/CASEID/restart directory. (Note this is the CASEID of the current run, not the original run).
 - 1. mkdir /ptmp/USER/archive/caseid/restart/
 - 2. untar the restart tar file, then place the tar file into a subdirectory while you test and load balance the run.
 - 3. keep a copy of the restart file in a safe place.
3. Edit env_mach.bluevista (set MSS: archiving, project, and path)
 - setenv DOUT_L_MS TRUE
 - setenv DOUT_L_MSNAME `echo \$LOGNAME | tr '[a-z]' '[A-Z]`
 - setenv DOUT_L_MSROOT /CCSM/csm/\$CASE
 - setenv DOUT_L_MSPWD \$DOUT_L_MSNAME
 - setenv DOUT_L_MSRPD 3650
 - setenv DOUT_L_MSPRJ xxxxxxxxx
4. configure the model: configure -mach bluefire
5. Edit env_run
 - STOP_OPTION
 - STOP_N
 - REST_OPTION
 - REST_N
6. Edit b30.myrun.bluefire.run
 - bsub -q regular
 - bsub -W 6:00
 - bsub -P PROJECT NUMBER
7. Edit b30.myrun.bluefire.l_archive
 - Change project number for MSS storage
 - After debugging, remove comment: ##BSUB -W 2:00
8. Add coupler mapping files, mksrf files, etc.
 (\$CASEROOT/SourceMods/src.[component])
9. Edit namelists (\$CASEROOT/Buildnml_prestage)



10. Build: `b30.myrun.bluefire.build`
11. Submit: `bsub < b30.myrun.bluefire.run`
12. Check standard output and standard error: `poe.stderr` and `poe.stdout`
13. Check `surface_dataset` (if newly created) for erroneous wetlands. Correct if necessary.



8.5 LSM vegetation types

No Vegetation

- 0 ocean
- 1 land ice
- 2 desert

Forest

- 3 cool needleleaf evergreen tree
- 4 cool needleleaf deciduous tree
- 5 cool broadleaf deciduous tree
- 6 cool mixed forest
- 7 warm needleleaf evergreen tree
- 8 warm broadleaf deciduous tree
- 9 warm mixed forest
- 10 tropical broadleaf evergreen forest
- 11 tropical broadleaf deciduous tree

Interrupted Woods

- 12 savanna
- 13 evergreen forest tundra
- 14 deciduous forest tundra
- 15 cool forest crop
- 16 warm forest crop

Non-woods

- 17 cool grassland
- 18 warm grassland
- 19 tundra
- 20 evergreen shrub land
- 21 deciduous shrub land
- 22 semi-desert
- 23 cool irrigated crop
- 24 cool crop
- 25 warm irrigated crop
- 26 warm crop

Wetland

- 27 forest wetland
- 28 non-forest wetland



8.6 CLM3 PFTs

PFT	Description
0	bare
1	needleleaf evergreen temperate tree
2	needleleaf evergreen boreal tree
3	needleleaf deciduous boreal tree
4	broadleaf evergreen tropical tree
5	broadleaf evergreen temperate tree
6	broadleaf deciduous tropical tree
7	broadleaf deciduous temperate tree
8	broadleaf deciduous boreal tree
9	broadleaf evergreen temperate shrub
10	broadleaf deciduous temperate shrub
11	broadleaf deciduous boreal shrub
12	arctic c3 grass
13	cool c3 grass
14	warm c4 grass
15	crop

8.7 QuickGuide: kmtEd – Deep Time

(Copied from include/controls.h)

Mouse Clicks:

- **shift + left mouse button**
allows user to change cell elevation (kmt)
- **ctrl + left mouse button**
changes connected cell group to default elevation (kmt)
- **middle mouse button**
zooms in and out quickly depending on mouse location
- **right mouse button**
moves the earth around in the window left to right and up and down

Key Strokes: (keyed in data viewing window)

'h' - rotates the view 30 degrees left

'j' - rotates the view 15 degrees down



'k' - rotates the view 15 degrees up
'l' - rotates the view 30 degrees right
't' - toggles topography visibility
'c' - toggles continental outline (coastline) visibility
'g' - toggles gridline visibility
'p' - saves image in PostScript format
'q' - quits and prompts for save option
'r' - returns view to original settings
'v' - saves high-resolution image in PostScript format
'z' - zooms to a user specified latitude and longitude
'<' - zooms in
'>' - zoom out



8.8 Startup near-modern glacial simulation

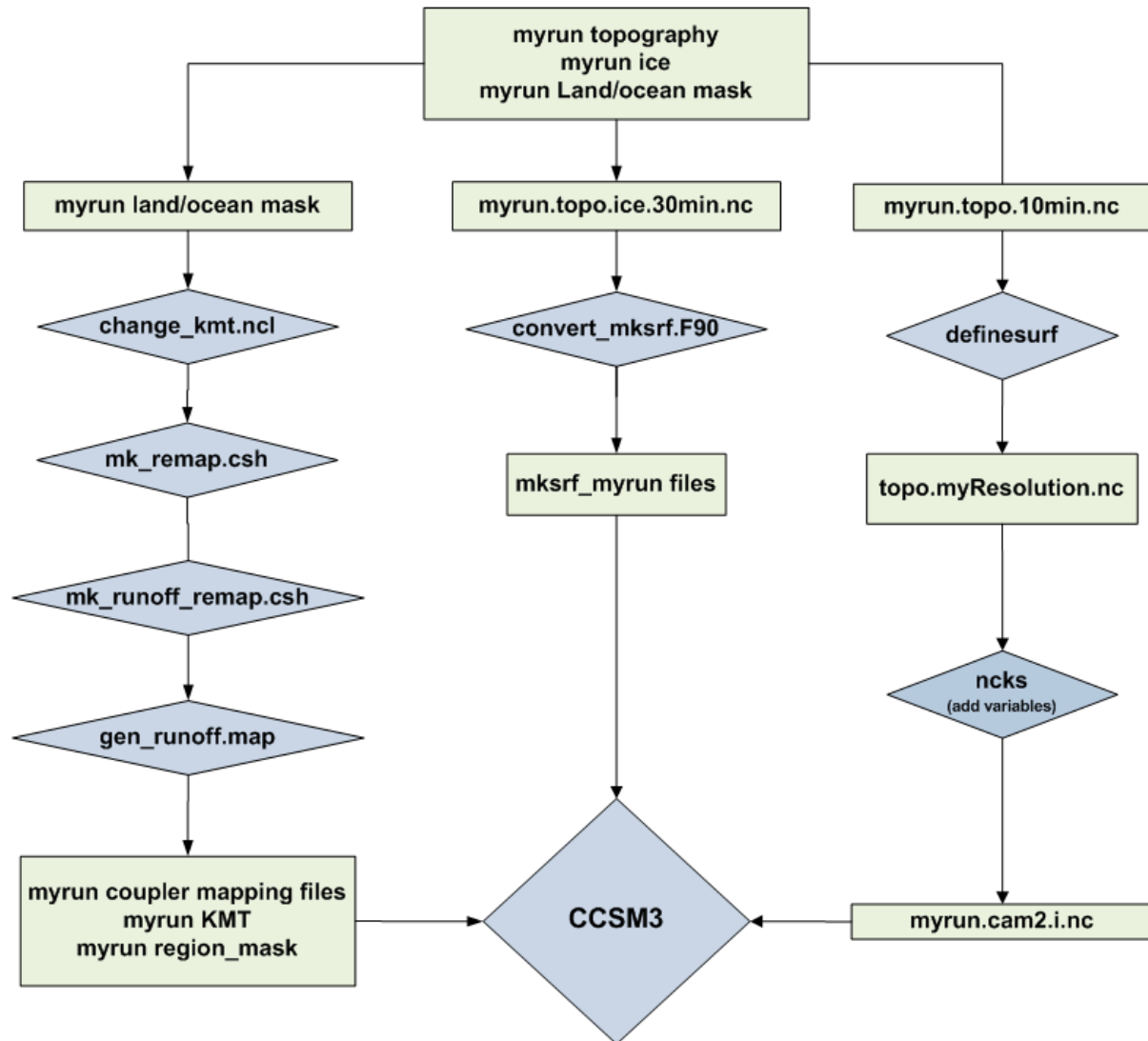


Figure 9: Startup glacial simulation

Schematic overview of the steps required to set up a hypothetical paleo simulation with minor changes to the present day land/ocean mask and the addition of 21ka land ice. Land model soil biophysics will start from initial conditions. Ocean and ice restart files can be used from the present day simulation (ref.pop.r and ref.csim.r) as long as no new ocean cells have been defined.



8.9 Hybrid near-modern glacial simulation

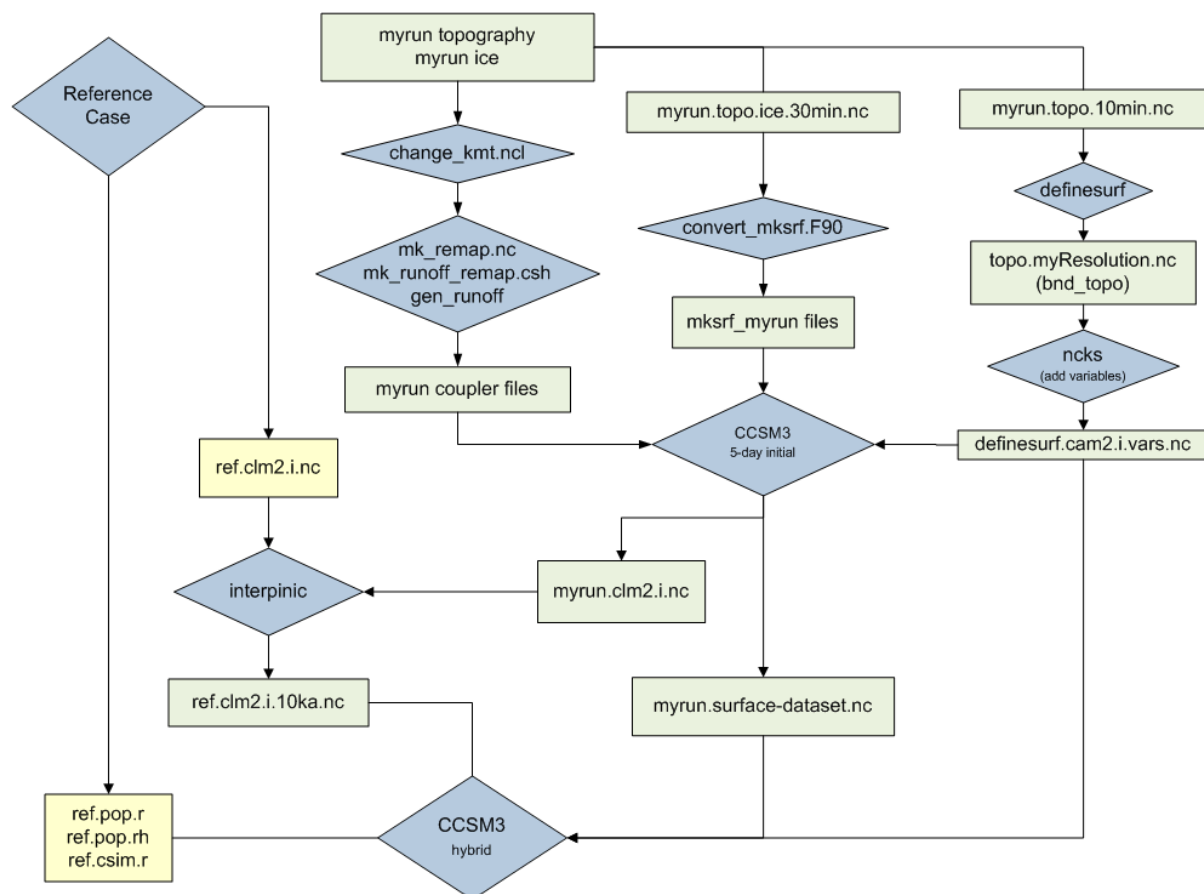


Figure 10: Hybrid glacial simulation

Schematic overview of the steps required for creating a hybrid 10ka simulation with (a) North American glaciers, and (b) land model soil biophysics initialized from an existing present day simulation. The tool *interpinic* projects the spunup soil conditions from the present day simulation (*ref.clm2.i.nc*) onto the 10ka landscape (*10ka.clm2.i.nc*). Ocean and ice restart files can be used from the present day simulation (*ref.pop.r* and *ref.csim.r*) as long as no new ocean cells have been defined.



8.10 Modifying the present day POP2 KMT

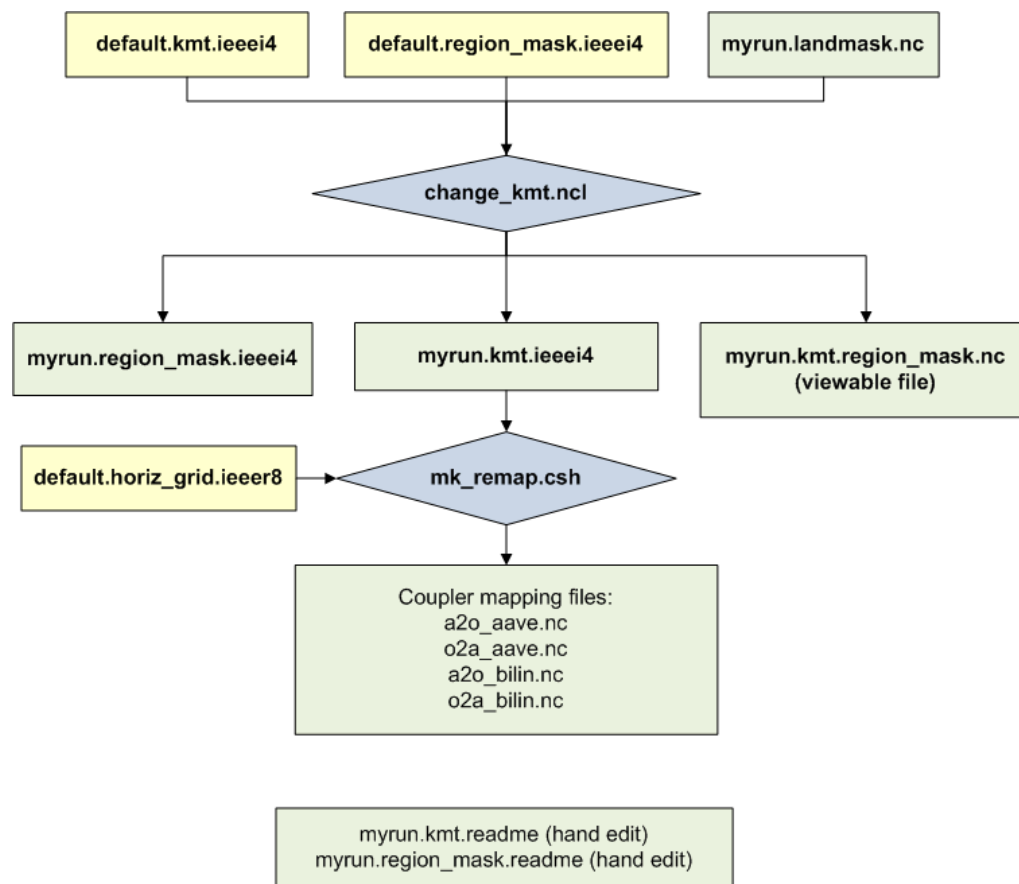


Figure 11: Changing the present day land/ocean mask.

This schematic shows the procedure for changing the present day ocean KMT in order to change the land/ocean mask (e.g., to cover Hudson Bay with land ice, or remove Hudson Bay for Pre-Quaternary simulations.). Without changing the KMT, Hudson Bay will remain defined as ocean in the fully coupled model, regardless of how you have defined it for the land model.



8.11 Coupler Mapping

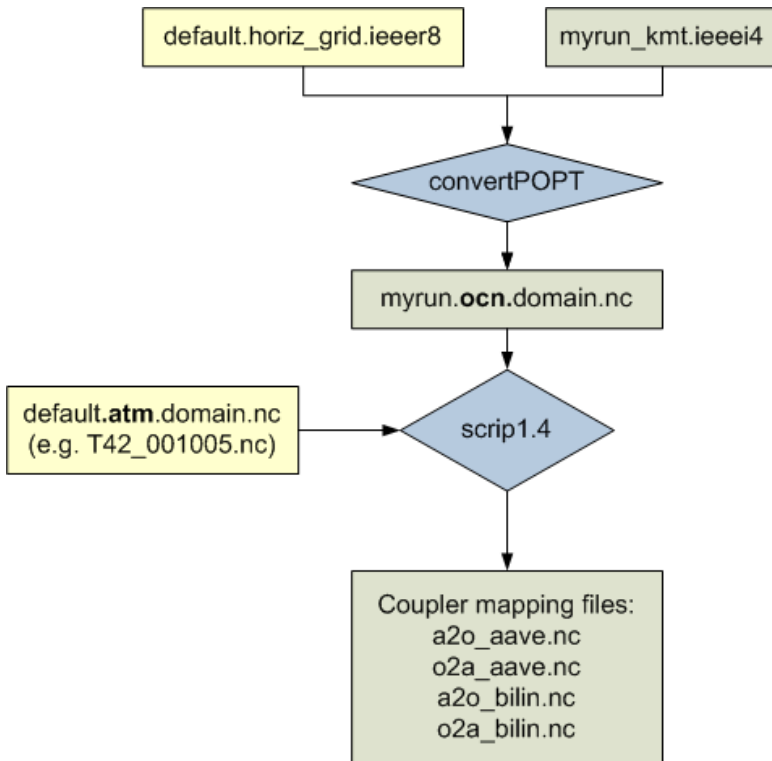


Figure 12: Coupler mapping.

Schematic shows the progression for creating new coupler mapping files. The mapping files direct fluxes between the ocean and atmosphere grids, and must be recreated when any changes are made to the default land/ocean mask.



8.12 River runoff mapping to the ocean grid

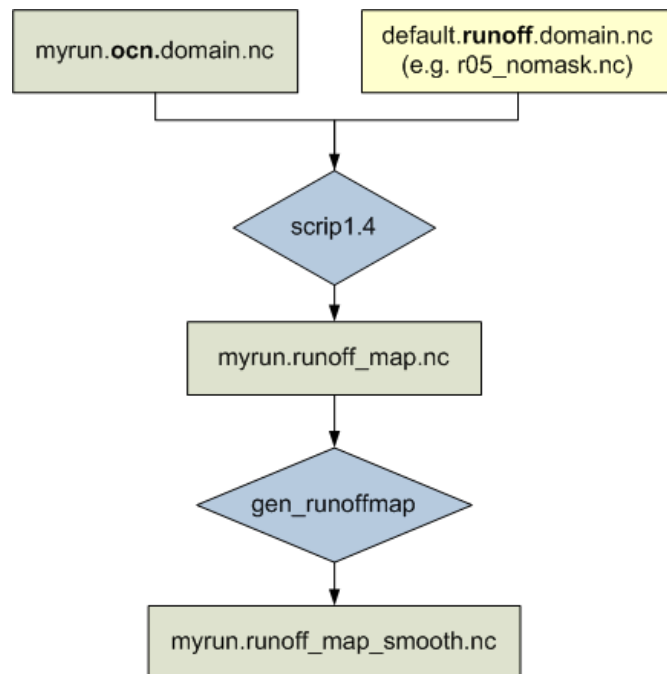


Figure 13: Mapping river runoff to the ocean.

Schematic shows the progression for creating coupler files for mapping river runoff onto the ocean grid. This process distributes river runoff onto ocean grid cells in a smooth halo around the river mouth. These files must be recreated when any changes are made to the default land/ocean mask.

8.13 References

- Berger, A., M-F. Loutre, C. Tricot, 1993: *Insolation and Earth's Orbital Periods. Journal of Geophysical Research. Vol. 98, No. D6, p.10,341-10,362.*
- Collins, W. et al., 2006. *The Community Climate System Model Version 3 (CCSM3). Journal of Climate. V19, No. 11, p.2122-2143.*
- Levitus, S., T. Boyer, M. Conkright, D. Johnson, T. O'Brien, J. Antonov, C. Stephens, and R. Gelfeld, 1998: *Introduction. Vol. 1, World Ocean Database 1998, NOAA Atlas NESDIS 18, 346 pp.*
- SCRIP: *A Spherical Coordinate Remapping and Interpolation Package. P. W. Jones. 1997. Los Alamos National Laboratory. <http://climate.lanl.gov/Software/SCRIP/>*
- Steele, M., R. Morley, and W. Ermold, 2001: *PHC: A global ocean hydrography with a high-quality Arctic Ocean. J. Climate, 14, 2079-2087.*



