PaleoToolkit Documentation for Deeptime CESM1.2

Includes:

A. Setup Recipe
B. Examples of Deep time Cases
C. “Gotcha’s” (Typical runtime issues)

Authors: Mathew Rothstein, Nan Rosenbloom, Christine Shields
Changelog: January 2, 2020

A. Setup Recipe

00. Create a workspace and cd to that directory

OCEAN Setup

01. Ocean: create ocean grid files (assumes a 1deg ocean)

input:

topography-bathymetry file	(0.5deg, 1deg or 2deg)
vertical grid	(for example, gx1v6_vert_grid)

output:

gridkmt.$ITER.nc	(used in step 5)
kmt.$ITER.da	(used in step 2, and user_nl_cice)
grid.$ITER.pop.da	(used in step 2,10,11, and user_nl_cice)
grid.$ITER.plot.da	(used in step 11)
h.$ITER.da	
output.$ITER.ns_dipole	(output log for run of ns_dipole)
output.$ITER.paleotopo	(output log for run of paleotopo)
output.$ITER.bin2nc	(output log for run of bin2nc)

01a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
               mk_ocn_grid/mk_grid_1x1_template.csh
```

Rename the template file to something appropriate for your work, for example, mk_grid_1x1_PETM.csh.

- set CASE (name appropriate for time period, ie PETM)
- set ITER (start at one and increment with each iteration)
- set topopath (full path to location of topo-bath file)
- set topofile (name of topo-bath file)
- set vrtgridpath (full path to location of vertical grid file)
- set vrtgrid (name of vertical grid file)
  note: find examples for vrtgrid in model code (~models/ocn/pop2/input_templates)
- set lonnp,latnp,lonsp,latsp (specifies location of poles)
- set jcon (# of rows of constant dy at poles, 11 is a place to start for 1deg)

01b.
Get cheyenne executables:

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
           mk_ocn_grid/ns_dipole
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
           mk_ocn_grid/paleotopo
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
           mk_ocn_grid/grid_bin2nc
```

Note: if using a machine other than cheyenne or executables require rebuild, get source code and makefile template to build new executables:

```
> svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
           mk_ocn_grid/mk_grid_src/
> cd mk_grid_src
> mv makefile.nwsc.bigEndian to 'makefile'
> make all
> mv ns_dipole paleotopo grid_bin2nc ../.
> cd ..
```

01c.

```
./mk_grid_1x1_$CASE.csh
```

02. Ocean: create SCRIP mapping file

input:
grid.$ITER.pop.da   (from step 1)
kmt.$ITER.da   (from step 1 or step 8)

output:

$ocnres_<date>.nc   (SCRIP mapping file, used in steps 3,13,25,26)

02a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
              mk_ocn_grid/mk_SCRIPgrid_template.csh
```

Rename the template file to something appropriate for your work, for example, mk_SCRIPgrid_PETM.csh

- set popgriddir to location of grid.$ITER.pop.da, kmt.$ITER.da from step 1
- set scripdir to location of myconvertPOPT executable (your current directory if downloading as described below)
- set popgrid, kmtgrid to names of output files from step 1
- set ocngridname to string appropriate for time period ('petm', for example)
- set ocnres to string appropriate for resolution and time period (ie gx1PETM)

02b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
              mk_ocn_grid/myconvertPOPT
```

Note: if using a machine other than cheyenne or executable requires rebuild, get source code and makefile template to build new executable:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
              mk_ocn_grid/mk_SCRIPgrid_src/
  > cd mk_SCRIPgrid_src
  > make myconvertPOPT
  > mv myconvertPOPT ../.
  > cd ..
```

02c.

```
./mk_SCRIPgrid_$CASE.csh       (make sure nco module is loaded)
```

03. Ocean: create regional diagnostic plots

input:
$ocnres_$date.nc   (from step 2)

output:
Grid_$ocnres_$date_reg_sp.ps
Grid_$ocnres_$date_reg_np.ps
Grid_$ocnres_$date_reg_q1.ps
Grid_$ocnres_$date_reg_q2.ps

```
Grid_$ocnres_$date_reg_q3.ps
Grid_$ocnres_$date_reg_q4.ps
```

03a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
        mk_ocn_grid/plot_all.sh
```

- set ocnres exactly as it was set in step 2 (ie gx1PETM)
- set date to match the date specified in output file of step 2 (yymmdd)

03b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
        mk_ocn_grid/plot_global_all.ncl
```

- no changes should be required

03c.

```
./plot_all.sh
```

(make sure ncl module is loaded)

04. Ocean: iterate between steps 1,2,3 until poles are in proper place

- use gv or some other tool to view *.ps files from step 3
- poles should be sufficiently inland - at least several grid cells away from coasts
- np and sp on same longitude may help
- instabilities in model run may require revisiting these steps

05. Ocean: optional KMT editor (GUI tool to modify lnd/ocn mask and ocn depths)

input:
gridkmt.$ITER.nc (from step 1 or step 6)

output:
gridkmt_fixed.nc (glitch in tool occassionally names this file "_fixed.nc")

05a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
        mk_ocn_grid/cesmGUITools/
```

05b. run the program

- make sure you login with Xwindows enabled - for example: 'ssh -Y -l <user> cheyenne.ucar.edu'
- this tool should be run on one of the data analysis nodes: 'execdav -a <project_number>'
- module load ncarenv
- module load python/2.7.16

- ncar_pylib
- setenv PYTHONPATH ./cesmGUITools/utilities
- ./cesmGUITools/editors/KMTEditor.py ./gridkmt.$ITER.nc
  note: the GUI tool will have instructions on how to edit file
- click File/Save Data if you want your changes saved (otherwise just
  close the GUI and decline the request to save your changes)
- type 'exit' to end usage of the analysis node

06. Ocean: rename saved file from step 5

input:
gridkmt_fixed.nc   (or "_fixed.nc", from step 5)

output:
gridkmt.$ITER.nc   (provide new iteration # in file name)

06a.

```
mv gridkmt_fixed.nc gridkmt.$ITER.nc
```

07. Ocean: check grid against an example ocn history file

input:
gridkmt.$ITER.nc   (from step 6)
gx1v6_ocn.nc   (example ocn history file)

output:
none

07a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
              mk_ocn_grid/gx1v6_ocn.nc
```

07b.

```
ncdump -v HTN,HTE gx1v6_ocn.nc
```

07c.

```
ncdump -v gridkmt.$ITER.nc
```

Note: variables HTN and HTE in gridkmt.$ITER.nc should be >= to those in gx1v6_ocn.nc. The units used for these variables may differ between files

08. Ocean: create new binary kmt file from netcdf file (if step 6 performed)

input:
gridkmt.$ITER.nc   (from step 6)

output:
kmt.$ITER.da   (increment the iteration)   (used in step 2, user_nl_cice)

08a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
            mk_ocn_grid/gridkmt_nc2bin
```

If using a machine other than cheyenne or executable requires rebuild,
get source code and makefile template to build new executable:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
              mk_ocn_grid/gridkmt_nc2bin_src/
  > cd gridkmt_nc2bin_src
  > make
  > mv gridkmt_nc2bin ../.
  > cd ..
```

08b.

```
./gridkmt_nc2bin  (interactive program that prompts for input and output filenames
```

09. Ocean: re-create the SCRIP mapping file   (if step 8 performed)

- repeat steps 2 and 3 with new kmt file, then continue to step 10

10. Ocean: ocn region mask and ocn transports

input:
kmt.$ITER.da   (from step 1 or 8)
grid.$ITER.pop.da   (from step 1)

output:
region.$CASE.be.ieeei4   (used in steps 11,12 and user_nl_pop2)
region_ids_$CASE   (used in user_nl_pop2)
transport_contents_$CASE   (used in user_nl_pop2)

10a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
            mk_ocninput/mk_ocninput_1x1_template.csh
```

Rename the template file to something appropriate for your work, for example, mk_ocninput_PETM.csh

- set CASE as in earlier steps
- set GRID to grid.$ITER.pop.da (from step 1)
- set KMT to kmt.$ITER.da (from step 1 or 8)
- customize section on region_ids (script will have some guidance)
- customize section on transport_contents (script will have some guidance)

10b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
               mk_ocninput/modregmsk_edit.f
```

- customize code to define ocean regions by i,j coordinates (must Be consistent with the region_ids specified in mk_ocninput script).

  Note: be sure to edit the *_edit.f and not the *.f file that results from executing mk_ocninput.csh

  Note: defining regions in modregmsk_edit.f can be tricky and it may help to use ncview on the gridkmt.$ITER.nc file from step 1 or step 8, choose variable kmt (change the range to 0-1 so that the land mask is easily seen) and then hover the cursor over the desired map points to get the x,y coordinates - these will roughly match the i,j coordinates that modregmsk_edit.f will require

10c.

```
./mk_ocninput.csh
```

Note: scan output for error messages!

11. Ocean: view region mask

input:

vertical grid file	(same as used in step 1)
region.$CASE.be.ieeei4	(from step 10)
grid.$ITER.plot.da	(from step 1)
grid.$ITER.pop.da	(from step 1)

output:
region_mask.nc

11a.

```
./grid_bin2nc      (same executable as used in step 1 - run interactively)
```

Note: interactive program expects these entries:
320 (lon dimension for 1deg ocean)
384 (lat dimension for 1deg ocean)
<choose name for output netcdf filename> (ie region_mask.nc)
depth profile filename (vertical grid file, as used in step 1)
KMT filename (region.$CASE.be.ieeei4, from step 10)
binary plot filename (grid.$ITER.plot.da, from step 1)
binary grid filename (grid.$ITER.pop.da, from step 1)

11b.
ncview region_mask.nc (make sure ncview module is loaded)

Note: Variable KMT will display the various ocean regions defined.
Go back to step 10 if regions require further modification

12. Ocean: check region mask

input:
region.$CASE.be.ieeei4 (from step 10)
kmt.$ITER.da (from step 1 or 8)

output:
last line of output to screen should be "Region mask and KMT match exactly"

12a.
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/mk_ocninput/cmpRegionMask2KMT.ncl
- set IPATH to location of latest kmt file (if different than current directory)
- set ifile1 to kmt.$ITER.da (from step 1 or 8)
- set ifile2 to region.$CASE.be.ieeei4 (from step 10)

12b.
ncl cmpRegionMask2KMT.ncl (make sure ncl module is loaded)

Coupler Setup

13. Coupler: coupler mapping

input:
$ocnres_$date.nc (SCRIP mapping file from step 2)
/glade/p/cesmdata/cseg/mapping/grids/<atm_grid_file>

note: grid files at various resolutions are available from same
directory - depending on intended resolution of atm/land (atmres):
fv1.9x2.5_090205.nc
fv0.9x1.25_141008.nc
fv0.23x0.31_141008.nc

output:
PET0.RegridWeightGen.Log (output log)
map_fv$atmres_TO_$ocnres_aave.$date.nc (used in env_run.xml, ATM2OCN_FMAPNAME)
map_fv$atmres_TO_$ocnres_blin.$date.nc (used in env_run.xml, ATM2OCN_SMAPNAME)
map_fv$atmres_TO_$ocnres_patc.$date.nc (used in env_run.xml, ATM2OCN_VMAPNAME)
map_$ocnres_TO_fv$atmres_aave.$date.nc (used in env_run.xml, OCN2ATM_FMAPNAME and step 14)
map_$ocnres_TO_fv$atmres_blin.$date.nc (doesn't appear to get used)

13a. create symbolic links:

```
ln -s <path_to_model_code>/tools/mapping/gen_mapping_files/gen_cesm_maps.sh .
ln -s <path_to_model_code>/tools/mapping/gen_mapping_files/gen_ESMF_mapping_file/ .
```

13b.

```
./gen_cesm_maps.sh -fatm /glade/p/cesmdata/cseg/mapping/grids/<atm_grid_file> \
-natm fv$atmres -focn $ocnres_$date.nc -nocn $ocnres --nogridcheck
```

14. Coupler: create domain files

input:
map_$ocnres_TO_fv$atmres_aave.$date.nc (from step 13)

output:
domain.lnd.fv$atmres_$ocnres.$date.nc (used in env_run.xml, step 32)
domain.ocn.$ocnres.$date.nc (used in env_run.xml)
domain.ocn.fv$atmres_$ocnres.$date.nc (used in env_run.xml, atm-only run)

14a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/
           cpl/gen_domain
```

note: if using a machine other than cheyenne or executables require rebuild,
get source code and makefile template to build new executable:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/
             cpl/gen_domain_src
  > cd gen_domain_src
  > make
  > cd ..  (gen_domain executable will be placed in parent directory)
```

14b.

```
./gen_domain -m map_$ocnres_TO_fv$atmres_aave.$date.nc -o $ocnres -l $atmres -c
<comment> -p 2
```

Note: '-p 2' required to ensure lats begin/end with 90.0,-90.0
Note: '-c' allows user to enter a comment as metadata in resulting netcdf files

Land Setup

15. Land: create the raw surface datasets

input:

```
mksrf_zon_organic.10level.nc                  (present-day soil values)
/glade/p/cesm/cseg/inputdata/lnd/clm2/rawdata/
     mksrf_soitex.10level.c010119.nc          (hard-coded in)
topography-bathymetry file                    (same as used in step 1)
LSM_datafile
```

Note: the code requires LSM values as input which are then converted to CLM-compatible plant functional types (pft's). If you don't have LSM values, you'll need to convert them - an example ncl script that converts from Biome4 can be grabbed here:

```
  svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
                landtype_convert_1deg.ncl
```

Note: LSM data and topo data should be at same resolution AND the datafiles should have longitude ordered 0to360 (use ncview or ncdump to verify) If data is -180to180, you'll need to reorder. For example, the following steps were used to adjust LSM data from Sewell, et al

```
  > cp Eoveg2x2ready.nc Eoveg2x2ready_0to360.nc
  > ncl
  > fn = "Eoveg2x2ready_0to360.nc"
  > f1 = addfile(fn,"rw")
  > mySUR = f1->SUR
  > mySUR = lonFlip(mySUR)
  > f1->SUR = mySUR
```

output:

```
mksrf_glacier_$CASE.c<date>.nc (used in step 17)
mksrf_urban_$CASE.c<date>.nc   (used in step 17)
mksrf_lanwat_$CASE.c<date>.nc  (used in steps 16,17)
mksrf_landuse_$CASE.c<date>.nc (used in step 17)
mksrf_lai_$CASE.c<date>.nc     (used in step 17)
mksrf_soicol_$CASE.c<date>.nc  (used in step 17)
mksrf_soitex_$CASE.c<date>.nc  (used in step 17)
mksrf_organic_$CASE.c<date>.nc (used in step 17)
mksrf_fmax_$CASE.c<date>.nc    (used in step 17)
mksrf_topo_$CASE.c<date>.nc    (used in step 17)
mksrf_vocef_$CASE.c<date>.nc   (used in step 17)
```

15a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            run_paleo_mkraw_cesm1_template.csh
```

Rename the template file to something appropriate for your work, for example, run_paleo_mkraw_cesm1_PETM.csh

- set CASE as in earlier steps
- set INPUT_LSM_DATA (name of input file containing land cover types)
- set INPUT_TOP_DATA (name of topo-bath file)

15b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            paleo_mkraw_cesm1_sed.F90
```

- set nlon,nlat to match resolution of topo and LSM inputs (for 1deg, nlon=360,nlat=180)
- expected var name for topo info in INPUT_TOP_DATA is 'topo', modify accordingly
- expected var name for pft info in INPUT_LSM_DATA is 'SUR', modify accordingly
- increase length of character arrays if LSM or topo filenames > 80chars
- review soil_color setting - default is 10 but an alternative may be appropriate for your setup - see Table 3.3 in CLM4 doc (http://www.cesm.ucar.edu/models/cesm1.0/clm/CLM4_Tech_Note.pdf)

15c.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            Makefile
```

- no changes should be required

15d.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
```

```
                mksrf_zon_organic.10level.nc
```

15e.

```
./run_paleo_mkraw_cesm1_$CASE.csh
```

Note: output to screen can fly by - watch for errors such as:

```
ERROR mapping veg to pct_pft: veg < 0 OR veg > 28  (invalid LSM value)
or
ERROR: sumpctpft =  0.000000000000000E+000        (should equal 100%)
```

16. Land: create the SCRIP grid

input:
mksrf_lanwat_$CASE.c<date>.nc   (from step 15)

output:
SCRIPgrid_$CASE_$nlatx$nlon.<date>.nc   (used in step 17)

16a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
                mkscripgrid_template.ncl
```

Rename the template file to something appropriate for your work, for example, mkscripgrid_PETM.ncl

- set name   (name of case as in earlier steps)
- set fn1   (name of lanwat file from step 15)
- set ipath (if different than current directory)

16b.

```
ncl mkscripgrid_$CASE.ncl        (make sure ncl module is loaded)
```

17. Land: create mksurfdata mapping file

input:
SCRIPgrid_PETM_180x360.<date>.nc (from step 16)

output:
map_<LSM_res>_$CASE_to_$atmres_nomask_aave_da_c<date>.nc   (used in step 18)
regridbatch_$CASE.o######   (output log)
PET###.RegridWeightGen.log   (output logs)

17a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            mkmapdata_paleo.sh
```

- set INGRID (SCRIPgrid file from step 16)
- set grids (string with LSM resolution and casename to be included in outfile name, for example, 1x1_PETM)

Note: modify the setting for grids under the section for clm4_0

17b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            regridbatch_paleo.sh
```

- set batch job charge account key
- set resols to desired output resolution (resolution of atm/lnd: for example, 1.9x2.5)
- set csmsrc to top level directory of model source code (directory containing subdirs "models" and "scripts")

17c.

```
qsub regridbatch_paleo.sh
```

18. Land: complete the land surface dataset

input:

map_<LSM_res>_$CASE_to_$atmres_nomask_aave_da_c<date>.nc	(from step 17)
mksrf_glacier_$CASE.c<date>.nc	(from step 15)
mksrf_urban_$CASE.c<date>.nc	(from step 15)
mksrf_lanwat_$CASE.c<date>.nc	(from step 15)
mksrf_landuse_$CASE.c<date>.nc	(from step 15)
mksrf_lai_$CASE.c<date>.nc	(from step 15)
mksrf_soicol_$CASE.c<date>.nc	(from step 15)
mksrf_soitex_$CASE.c<date>.nc	(from step 15)
mksrf_organic_$CASE.c<date>.nc	(from step 15)
mksrf_fmax_$CASE.c<date>.nc	(from step 15)
mksrf_topo_$CASE.c<date>.nc	(from step 15)
mksrf_vocef_$CASE.c<date>.nc	(from step 15)

output:

```
fsurdat       = 'surfdata_$atmres_$CASE_c<date>.nc'   (used in user_nl_clm)
fsurlog       = 'surfdata_$atmres_$CASE_c<date>.log'
mksrf_fdynuse  = 'pftdyn_hist_$CASE.txt'
fdyndat       = 'surfdata.pftdyn_$atmres_$CASE_c<date>.nc'
```

18a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            mksurfdata_map
```

Note: if using a machine other than cheyenne or executables require rebuild, get source code and makefile to build new executable:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/
              lnd/mksurfdata_map_src
  > cd mksurfdata_map_src/src
  > make
  > cd ../..
  > mv mksurfdata_map_src/mksurfdata_map .
```

18b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/
            mksurfdata_map.namelist.paleo
```

- set mapping_file to mapping file from step 17
- set mksrf_* to mksrf_* files from step 15
- set out_res,casename and date in output filenames   (out_res = $atmres)

18c.

```
./mksurfdata_map < mksurfdata_map.namelist.paleo
```

Note: it is strongly recommended to view the new surface dataset just created to make sure all values are as expected - in particular, any PFT related vars

Runoff Setup

19. Runoff: prep topo file at proper resolution

To help the process of making changes in the runoff setup, it is highly recommended to run RTM (the active runoff model required for deep-time model runs in CESM) at a resolution no finer than 1deg. If the topography-bathymetry file used in earlier steps needs to be interpolated to another resolution, the following tool may help.

input:
topography-bathymetry file   (same as used in step 1)

output:
topo.1x1deg.$CASE.nc   (used in step 20)

19a.
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
              create-topo_1x1deg.ncl

Note: code for other resolutions are available from same directory

- set CASE as in earlier steps
- set fili to topo-bath file to interpolate from
- code assumes variables in topo file are named 'lat','lon', and 'topo'
  modify variable names in ncl code accordingly

19b.
ncl create-topo_1x1deg.ncl   (make sure ncl module is loaded)

20. Runoff: generate runoff data from topo inputs

input:
topography/bathymetry file (from step 19 or same file as was used in step 1)

output:
fort.10_$CASE (CLM-required format, before inf loops fixed, used in step 21)
fort.11_$CASE (list of points involved in inf loops, used in step 21)
fort.12_$CASE
fort.13_$CASE (CLM-required format, after inf loops fixed, used in step 21)
fort.14_$CASE
fort.15_$CASE

20a.
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
              rdirc_template.csh

Rename the template file to something appropriate for your work, for example, rdirc_PETM.csh

- set CASE as in earlier steps
- set INFILE to path and name of input topo/bath file (from step 19 or same
  file as was used in step 1)

20b.
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/

```
            topo2rdirc_sed.F90
```

- code assumes variables in topo file are named 'lat','lon', and 'topo'
  modify variable names in fortran code accordingly
- modify nlon, nlat depending on desired resolution of output for RTM
  to run (which must match the resolution of the topo input)
  (for 1deg, nlon=360 and nlat=180)

20c.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
 Makefile
```

- no changes should be required

20d.

```
./rdirc_$CASE.csh
```

Note: Job may pause (reporting an error) when an issue is found (like an infinite loop). Just hit enter at the prompt and the program will attempt to correct the issue and the output files will document what was done.

21. Runoff: plot runoff (optional)

input:
topo-bath filename from step 19 (used in plot and output filename only)
fort.10_$CASE (from step 20)
fort.11_$CASE (from step 20)
fort.13_$CASE (from step 20)

output:
rdirc_<topo-bath_filename>.ps (uses input filename for output filename)

21a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
            plotrdirc.csh
```

- set IFILE (same topo-bath file used in step 20)
- set CASE (same casename as used in step 20)
- modify NLAT,NLON,RESOLN depending on resolution of input topo file

21b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
            plot_rdirc.ncl
```

- modify res@mpMinLatF, res@mpMaxLatF, res@mpMinLonF, res@mpMaxLonF in the section labeled "Zoom in on data" to zoom in on geographical area of interest (leaving these commented out will give global a plot)

21c.

```
./plotrdirc.csh         (make sure ncl module is loaded)
```

21d.

```
gv rdirc_<topo-bath_filename>.ps    (use ghostview or other tool to view plot)
```

Note: top plot is the uncorrected runoff map, bottom plot is the corrected; arrows are plotted showing direction of runoff at each grid cell

22. Runoff: modify runoff

input:
fort.13_$CASE   (from step 20)

output:
fort.13_$CASE

22a.

```
edit fort.13_$CASE     (use your favorite editor)
```

Note: fort.13_$CASE *should* have all infinite loops corrected, but if a change is needed to the runoff in a region (perhaps there are salinity issues in the model run) then runoff values in this file would need to be modified. This is a text file with three columns of data for lat,lon and direction, where values range from 0-8 (0=ocean, 1=north and rotating clockwise to 8=northwest).

23. Runoff: check for infinite loops (optional)

note: If fort.13_$CASE was modified in step 22, another check for any infinite loops introduced might be warranted; then iterate with steps 21 and 22 until satisfied

input:
fort.13_$CASE (from step 20 or 22)

output:
fort.11   (only produced if any infinite loops found)

23a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
              check_inf_loop.F90
```

- modify nlat,nlon for resolution of input
- set filei to 'fort.13_$CASE'

23b.

```
make check_inf_loop  (using same Makefile as step 20)
```

23c.

```
./check_inf_loop
```

Note: job may pause as issues are encountered and recorded - just hit enter at the prompt to continue. If the job does not pause, it likely encounted no infinite loops. Otherwise, just hit enter at the prompt to continue and any issues should be recorded in fort.11.

24. Runoff: convert runoff file to netcdf

input:
fort.13_$CASE (from step 20 or 22)

output:
rdirc.1x1.$CASE.nc (used in user_nl_rtm)

24a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
              rtm_ncdf.pro
```

- set rtmfile1 = 'fort.13_$CASE'
- set outfile = 'rdirc.1x1.$CASE.nc' ;approprate resolution in name
- set resnum=2 (1deg) or resnum=1 (0.5deg)
- set history attribute with proper documentation for user and date
- set source attribute with proper input file (fort.13_$CASE)

24b.
make sure you login with Xwindows enabled - for example:

```
'ssh -Y -l <user> cheyenne.ucar.edu'
module load idl
idl
IDL> .rn rtm_ncdf
IDL> rtm
IDL> exit
```

25. Runoff: create runoff to ocean mapping file

input:
fort.13_$CASE (from step 20 or 22)
$ocnres_$date.nc (SCRIP mapping file from step 2)

output:
map_$rofres$CASE_to_$ocnres_nn_<date>.nc
map_$rofres$CASE_to_$ocnres_sm_e1000r300_<date>.nc
map_$rofres$CASE_to_$ocnres_nnsm_e1000r300_<date>.nc (ROF2OCN_RMAPNAME in env_run.xml)

25a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
          runoff_map_1deg
```

Note: runoff_map_0.5deg executable is also available

Note: if using a machine other than cheyenne or executables require rebuild,
get source code and makefile template to build new executables:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
          runoff_map_src
  > cd runoff_map_src
  > cp src/map_mod_1deg.F90_save src/map_mod.F90  (or map_mod_0.5deg.F90_save)
  > ./build.cheyenne
  > cp runoff_map ../runoff_map_1deg            (or runoff_map_0.5deg)
  > cd ..
```

25b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
          runoff_map.1x1.template.nml
```

Rename the template file to something appropriate for your work, for example, runoff_map.1x1.PETM.nml

- set file_roff to fort.13_$CASE
- set file_ocn to the SCRIP_mapping file
- replace <casename> with $CASE in each of the output files and title
- replace <ocnres> with $ocnres in each of the output files and title (ie gx1PETM)
- replace <date> with current date in each of the output files

25c.

- create softlink to namelist file

```
ln -s runoff_map.1x1.$CASE.nml ./runoff_map.nml
```

25d.
grab an analysis node with the command, 'execdav -a <project_number>'

25e.

```
./runoff_map_1deg
```

25f.
- type 'exit' to end usage of the analysis node

26. Runoff: create runoff to ocean mapping file (part 2)

input:

```
$ocnres_$date.nc                    (SCRIP mapping file from step 2)
/glade/p/cesmdata/cseg/mapping/grids/1x1d.nc
```

Note: grids at 0.5deg (r05_nomask_070925.nc) and 2deg (r19.nc) are
also available from same grids directory, depending on runoff resolution

output:

```
map_r1_nomask_TO_$ocnres_aave.<date>.nc   (ROF2OCN_FMAPNAME in env_run.xml)
```

26a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
           create_ESMF_map.sh
```

- no changes should be required

26b.

```
./create_ESMF_map.sh -fsrc /glade/p/cesm/cseg/mapping/grids/1x1d.nc -nsrc r1_nomask -fdst ./$ocnres_$date.nc -ndst $ocnres -map aave
```

27. Runoff: create runoff to/from land mapping files - needed if rof at 1deg rather than 0.5 deg

Note: If running rtm (runoff model) at a resolotion other than 0.5 deg (it's default),
you'll need to specify mapping files between land and runoff that are specific to
your resolution. Many such maps can be found here:

```
/glade/p/cesmdata/cseg/inputdata/lnd/clm2/mappingdata/maps
```

If no mapping files exist for your resolutions (lnd and rof), use the following steps to create them.

input:
/glade/p/cesmdata/cseg/mapping/grids/1x1d_lonshift.nc
/glade/p/cesmdata/cseg/mapping/grids/fv$atmres_<date>.nc

Note: grids at other resolutions are available from same grids directory,
depending on resolution of atm/land

output:
map_r19_nomask_TO_r1x1_aave.<date>.nc (LND2ROF_FMAPNAME in env_run.xml)
map_r1x1_TO_r19_aave.<date>.nc (ROF2LND_FMAPNAME & ROF2LND_SMAPNAME in env_run.xml)

27a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/
          create_ESMF_map.sh
```

- same script as was used in step 26
- no changes should be required

27b.

```
./create_ESMF_map.sh -fsrc /glade/p/cesm/cseg/mapping/grids/fv$atmres_<date>.nc -nsrc r19_nomask -fdst /glade/p/cesm/cseg/mapping/grids/1x1d_lonshift.nc -ndst r1x1 -map aave
```

27c.

```
./create_ESMF_map.sh -fsrc /glade/p/cesm/cseg/mapping/grids/1x1d_lonshift.nc -nsrc r1x1 -fdst /glade/p/cesm/cseg/mapping/grids/fv$atmres_<date>.nc -ndst r19 -map aave
```

Atmosphere setup

28. Atmosphere: create a 10min topographic file

input:
USGS-gtopo30_10min_c050419.nc (already specified and pointed to in ncl script)
topography/bathymetry file (same file as was used in step 1)

Note: 2deg topo data probably will have interpolation problems - see step 19
for how to interpolate to a finer resolution, if necessary (1deg should work)

output:
$CASE_10min_topo_4input2definesurf.<date>.nc (used in step 29)

28a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            mk_10min_definesurf_input_paleo.ncl
```

- expected var name for topo info in topo file is 'topo', modify accordingly
- set cases with same casename as earlier steps
- set path to location of topo file
- set topoinput to name of topo-bath file

28b.

```
ncl mk_10min_definesurf_input_paleo.ncl      (make sure ncl module is loaded)
```

29. Atmosphere: create boundary dataset for topography fields

input:
$CASE_10min_topo_4input2definesurf.<date>.nc (from step 28)
landm_coslat.nc (used as template)
fv_$atmres.nc (grid map)

output:
bnd_topo_$CASE_$atmres_remap.<date>.nc

29a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            definesurf
```

Note: if using a machine other than cheyenne or executables require rebuild,
get source code and makefile to build new executable:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
              definesurf_src
  > cd definesurf_src
  > make
  > cp definesurf ../.
  > cd ..
```

29b.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            fv_$atmres.nc
```

Note: other grid maps available, depending on resolution of atm ie
fv_1.9x2.5.nc, fv_0.9x1.25.nc, fv_0.23x0.31.nc

29c.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            landm_coslat.nc
```

29d.

```
./definesurf -remap -r -t $CASE_10min_topo_4input2definesurf.<date>.nc -g fv_$atmres -l
landm_coslat.nc bnd_topo_$CASE_$atmres_remap.<date>.nc
```

30. Atmosphere: add cam5 required variables to boundary topography dataset

Note: cam5 requires standard deviation of geopotential height (SGH) as well as smoothed fractional land values (landm_coslat)

input:
bnd_topo_$CASE_$atmres_remap.<date>.nc (from step 29)

output:
bnd_topo_$CASE_$atmres_remap_sgh30.<date>.nc (to be used in user_nl_cam)

30a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            add_SGH30_paleo.ncl
```

- set cases with same casename as earlier steps
- set ifile1 to name of bnd_topo file from step 29
- set ofile to name of output file

30b.

```
ncl add_SGH30_paleo.ncl
```

(make sure ncl module is loaded)

31. Atmosphere: create solar forcing file

input:
/glade/p/cesmdata/cseg/inputdata/atm/cam/solar/SOLAR_SPECTRAL_Lean_1610-2008_annual_c090324.nc
(hard-coded into ncl script)

output:
solar_scon_$CASE.<date>.nc (used in user_nl_cam)

31a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
```

sol_constant_paleo.ncl

rename the template file to something appropriate for your work, for example, sol_constant_PETM.ncl

- set outfil with casename as in earlier steps and current date
- set So_adj to value appropriate for your model run

31b.
ncl sol_constant_$CASE.ncl   (make sure ncl module is loaded)

Note: this will modify the values at year 1850 in the output file - the model run looks for this year by default

32. Atmosphere: customize aerosol settings

input:
1850 aerosol files at $atmres   (used as starting point - listed in code)
topography-bathymetry file   (same as used in step 1)
domain.lnd.fv$atmres_$ocnres.$date.nc   (from step 14)
cam initial file   (from short run of the model)

Note: This cam initial file can be produced with a short run of your model setup without aerosols finalized. You can proceed to step 34 and run the model for just one year with the cam namelist setting, inithist = 'YEARLY' and then return to this step to complete your aerosol settings with the *cam.i* file from the model run directory.

output:
oxid_$atmres_L26_1850clim_c091123_for_$CASE.nc
ar5_mam3_so2_elev_1850_c090726_for_$CASE.nc
ar5_mam3_bc_elev_1850_c090726_for_$CASE.nc
ar5_mam3_num_a1_elev_1850_c090726_for_$CASE.nc
ar5_mam3_num_a2_elev_1850_c090726_for_$CASE.nc
ar5_mam3_oc_elev_1850_c090726_for_$CASE.nc
ar5_mam3_so4_a1_elev_1850_c090726_for_$CASE.nc
ar5_mam3_so4_a2_elev_1850_c090726_for_$CASE.nc
aerocom_mam3_dms_surf_2000_c090129_for_$CASE.nc
ar5_mam3_so2_surf_1850_c090726_for_$CASE.nc
ar5_mam3_soag_1.5_surf_1850_c100217_for_$CASE.nc
ar5_mam3_bc_surf_1850_c090726_for_$CASE.nc
ar5_mam3_num_a1_surf_1850_c090726_for_$CASE.nc
ar5_mam3_num_a2_surf_1850_c090726_for_$CASE.nc

```
ar5_mam3_oc_surf_1850_c090726_for_$CASE.nc
ar5_mam3_so4_a1_surf_1850_c090726_for_$CASE.nc
ar5_mam3_so4_a2_surf_1850_c090726_for_$CASE.nc
clim_p_trop_for_$CASE.nc
dst_$atmres_c090203_for_$CASE.nc
regrid_vegetation_for_$CASE.nc
```

Note: These are "paleo-tized" files in that the 1850 input values were zonally averaged and then tied to the geography of your time period. Use ncview to verify the land mask in the dst_* file, for example. The 1850 values can now more easily be swapped for values appropriate for your time period.

32a.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            modify_aerosol_input_template.ncl
```

Rename the template file to something appropriate for your work, for example, modify_aerosol_input_PETM.ncl

- set Pfil (two settings!) to lnd domain file from step 14
- set Fp to a cam initial file
- set top_in to your topography-bathymetry file
- expected var name for topo info is 'topo', modify accordingly
- set output_stem to intended path of output

33. Atmosphere: period-specific aerosols (OPTIONAL)

This will be unique to your setup. Below, we will give you an example of what we did for a PETM run.

33a. Go to step 34 to run the model for a longer time period (We used 20 years).
- set histaux_a2x3hr = .true. to produce cpl hist files

33b. Then do a land-only run (compset I1850SPINUPCN) to produce MEGAN variables (~2yrs)
To output MEGAN variables, you will need these additional settings:

in env_run.xml (you can use xmlchange):
- set DATM_MODE to CPLHIST3HrWx
- set DATM_CPLHIST_CASE to casename of run that produced cpl hist files
- set DATM_CPLHIST_YR_START to start year of cpl hist files produced
- set DATM_CPLHIST_YR_END to last year of cpl hist files produced

in user_nl_clm:

- set megan_specifier = 'ISOP = isoprene','C10H16 = myrcene + sabinene + limonene + ocimene_t_b + pinene_b + 2met_styrene + cymene_p + cymene_o + phellandrene_a + terpinene_a + terpinene_g + terpinolene + phellandrene_b + camphene + bornene + fenchene_a + ocimene_al + ocimene_c_b + carene_3 + pinene_a + thujene_a'

Note: Make sure setting for "megan_specifier" (from user_nl_clm) appears in ~/run/drv_flds_in. (Potential "gotcha", you may need to explicitly set this in your run directory).

33c. Take MEGAN output and replace previous paleotized values created from step 32 (noted as step 32 output).
(Instructions forthcoming, email shields@ucar.edu for more information).

B. Examples of Deep time Cases

Examples of how to set up a model run with a deep time compset

34. FULLY COUPLED:
Setting up a fully-coupled (B case) model run at 2deg atmosphere:

34a.

```
cd to new workspace
svn export https://svn-ccsm-models.cgd.ucar.edu/cesm1/exp_tags/pcesm_cesm1_2_2_tags/cesm-dt2.0_cesm1_2_2_1
```

34b.

```
cd cesm-dt2.0_cesm1_2_2_1/scripts
```

34c.

```
./create_newcase -case /<case_directory>/<casename>   \
            -res f19_g16                  \
            -mach cheyenne                  \
            -compset BPETMC5CN              \
            -user_mods_dir  ../usermods_dirs/BPETMC5CN
```

Note: This example creates the setup for a fully-coupled PETM run with the atmosphere (CAM5) at 2 degree. However, this can be used as a template for a run of any paleo period by using the tools in this guide and following the steps that follow. Experienced users will notice that many of the cesm setup files are automatically populated in the case directory and only require modification when customizing for your specific time period.

34d.

```
cd <case_directory>/<casename>
```

34e.

The default setup is to run as a 'startup' where the underlying component models will begin with arbitrary initial conditions. Alternatively, you could start as either a 'branch' or 'hybrid' with restart files from the end of a community 2000yr PETM run (available on Cheyenne's inputdata):

```
./xmlchange RUN_REFCASE="B_PETM_2deg_8x_aero.21"
./xmlchange RUN_REFDATE="2001-01-01"
./xmlchange GET_REFCASE="TRUE"
```

in user_nl_cice: comment out setting for ice_ic

additional setup for a branch:

```
./xmlchange RUN_TYPE="branch"
```

in user_nl_pop2:
comment out settings under "INITIAL RUNS ONLY" for init_ts_option and init_ts_file
uncomment out setting under "CONTINUE RUNS ONLY" for init_ts_option and set to 'ccsm_continue'

additional setup for a hybrid:

```
./xmlchange RUN_TYPE="hybrid"
```

comment out settings under "INITIAL RUNS ONLY" for init_ts_option and init_ts_file
uncomment out setting under "CONTINUE RUNS ONLY" for init_ts_option and set to 'ccsm_hybrid'

34f.

The variable CCSM_CO2_PPMV has the default setting for CO2 (currently set for the PETM).
This can be changed with:

```
./xmlchange CCSM_CO2_PPMV="<value>"
```

34g.

use xmlchange for any new domain settings with files created in step 14

```
./xmlchange ATM_DOMAIN_FILE="domain.lnd.fv19_25_$ocnres.$date.nc"
./xmlchange ATM_DOMAIN_PATH="<path_to_domain_file>"
./xmlchange LND_DOMAIN_FILE="domain.lnd.fv19_25_$ocnres.$date.nc"
./xmlchange LND_DOMAIN_PATH="<path_to_domain_file>"
./xmlchange ICE_DOMAIN_FILE="domain.ocn.$ocnres.$date.nc"
./xmlchange ICE_DOMAIN_PATH="<path_to_domain_file>"
./xmlchange OCN_DOMAIN_FILE="domain.ocn.$ocnres.$date.nc"
./xmlchange OCN_DOMAIN_PATH="<path_to_domain_file>"
```

34h.

use xmlchange for any new mapping files from step 13

```
./xmlchange ATM2OCN_FMAPNAME="<path>/map_fv19_25_TO_$ocnres_aave.$date.nc"
./xmlchange ATM2OCN_SMAPNAME="<path>/map_fv19_25_TO_$ocnres_blin.$date.nc"
./xmlchange ATM2OCN_VMAPNAME="<path>/map_fv19_25_TO_$ocnres_patc.$date.nc"
./xmlchange OCN2ATM_FMAPNAME="<path>/map_$ocnres_TO_fv19_25_aave.$date.nc"
./xmlchange OCN2ATM_SMAPNAME="<path>/map_$ocnres_TO_fv19_25_aave.$date.nc"
```

34i.

use xmlchange for any new mapping files from step 27

```
./xmlchange ROF2LND_FMAPNAME="<path>/map_r1x1_TO_r19_aave.<date>.nc"
./xmlchange ROF2LND_SMAPNAME="<path>/map_r1x1_TO_r19_aave.<date>.nc"
./xmlchange LND2ROF_FMAPNAME="<path>/map_r19_nomask_TO_r1x1_aave.<date>.nc"
```

34j.

use xmlchange for any new mapping file from step 26

```
./xmlchange
ROF2OCN_FMAPNAME="<path>/map_r1_nomask_TO_$ocnres_aave.<date>.nc"
```

34k.

use xmlchange for any new mapping file from step 25

```
./xmlchange
ROF2OCN_RMAPNAME="<path>/map_$rofres$CASE_to_$ocnres_nnsm_e1000r300_<date>.nc
"
```

34l.

- customize user_nl_cam
  - set bnd_topo with file from step 30
  - set solar_data_file with file from step 31
  - set tropopause_climo_file with file from step 32
  - set soil_erod with file from step 32
  - set tracer_cnst_datapath and tracer_cnst_file with file from step 32
  - set depvel_lnd_file with file from step 32
  - set ext_frc_specifier (SO2,so4_a1,so4_a2,num_a1,num_a2,bc_a1,pom_a1) with files from step 32
  - set srf_emis_specifier (SO2,so4_a1,so4_a2,num_a1,num_a2,bc_a1,SOAG,DMS,pom_a1) with files from step 32
  - customize other settings not addressed in this guide n2ovmr,ch4vmr,f11vmr,f12vmr
  - customize output settings fincl,nhtfrq,mfilt

34m.

- modify user_nl_clm with customized settings from following steps in this guide
  - set fsurdat with file from step 18

- customize output settings
  hist_fincl,hist_nhtfrq,hist_mfilt

34n.
- modify user_nl_cice with customized settings from following steps in this guide
  - set grid_file with grid.$iter.pop.da from step 1
  - set kmt_file with kmt.$iter.da from step 1 or 8
  - Note: by default, we initialize with zero ice (For periods that have ice, the ice state will spin up very quickly on it's own).

34o.
- modify user_nl_cpl with customized settings from following steps in this guide
  - adjust orbital settings, if necessary

34p.
- modify user_nl_pop2 with customized settings from following steps in this guide
  - set horiz_grid_file with grid.$iter.pop.da from step 1
  - set topography_file with kmt.$iter.da from step 1 or 8
  - set region_mask_file with region.$CASE.be.ieeei4 from step 10
  - set region_info_file with region_ids_$CASE from step 10
  - set diag_transport_file with transport_contents_$CASE from step 10
  - Note: You will notice that other physical settings necessary for paleo periods are already set by default.

34q.
- modify user_nl_rtm with customized settings from following steps in this guide
  - set frivinp_rtm with file from step 24
  - customize output settings
    rtmhist_fincl,rtmhist_nhtfrq,rtmhist_mfilt

34r.

```
modify pe layout in env_mach_pes.xml  (if necessary)
./cesm_setup
./<casename>.build
```

34s.

```
modify run settings in env_run.xml (length of run, etc)
modify run settings in <casename>.run  (charge account, job queue, etc)
./<casename>.submit
```

35. ATMOSPHERE-ONLY and HIGH RESOLUTION:

Setting up an atmosphere-only (F case) model run at 1/4deg atmosphere:

35a. (if not already performed)

cd to new workspace

```
svn export https://svn-ccsm-models.cgd.ucar.edu/cesm1/exp_tags/pcesm_cesm1_2_2_tags/
cesm-dt2.0_cesm1_2_2_1
```

35b.

```
cd cesm-dt2.0_cesm1_2_2_1/scripts
```

35c.

```
./create_newcase -case /<case_directory>/<casename>
            -res f02_f02
            -mach cheyenne
            -compset FPETMC5
            -user_mods_dir  ../usermods_dirs/FPETMC5
```

Note: This example creates the setup for an atmosphere-only PETM run with the atmosphere (CAM5) at 1/4 degree. However, this can be used as a template for a run of any paleo period by using the tools in this guide and following the steps that follow. Experienced users will notice that many of the cesm setup files are automatically populated in the case directory and only require modification when customizing for your specific time period or resolution.

35d.

```
cd <case_directory>/<casename>
```

35e.

The variable CCSM_CO2_PPMV has the default setting for CO2 (currently set for the PETM). This can be changed with:

```
./xmlchange CCSM_CO2_PPMV="<value>"
```

35f.

perform steps 13 and 14 using 0.23x0.31 as $atmres

use xmlchange for any new domain settings

```
  ./xmlchange ATM_DOMAIN_FILE="domain.lnd.fv0.23x0.31_$ocnres.$date.nc"
  ./xmlchange ATM_DOMAIN_PATH="<path_to_domain_file>"
  ./xmlchange LND_DOMAIN_FILE="domain.lnd.fv0.23x0.31_$ocnres.$date.nc"
  ./xmlchange LND_DOMAIN_PATH="<path_to_domain_file>"
  ./xmlchange ICE_DOMAIN_FILE="domain.ocn.0.23x0.31_$ocnres.$date.nc"
  ./xmlchange ICE_DOMAIN_PATH="<path_to_domain_file>"
  ./xmlchange OCN_DOMAIN_FILE="domain.ocn.0.23x0.31_$ocnres.$date.nc"
  ./xmlchange OCN_DOMAIN_PATH="<path_to_domain_file>"
  ./xmlchange
SSTICE_GRID_FILENAME="<path_to_domain_file>/domain.ocn.0.23x0.31_$ocnres.$date.nc"
```

35g.
perform step 27 using 0.23x0.31 as $atmres
use xmlchange for any new mapping files

```
./xmlchange ROF2LND_FMAPNAME="<path>/map_r1x1_TO_r02_aave.<date>.nc"
./xmlchange ROF2LND_SMAPNAME="<path>/map_r1x1_TO_r02_aave.<date>.nc"
./xmlchange LND2ROF_FMAPNAME="<path>/map_r02_nomask_TO_r1x1_aave.<date>.nc"
```

35h. create sst file from climo files of fully-coupled run with spun-up ocean

input:
atm climotology files from an existing fully-coupled run (~50yrs)

/glade/p/cesmdata/cseg/inputdata/ocn/docn7/SSTDATA/sst_HadOIBl_bc_0.23x0.31_clim_c061106.nc

Note: other files available, depending on your resolution

output:
sst_$atmres_$CASE_c<date>.nc   (used in env_run.xml and user_nl_cice)

35h_i. We will be using NCO operators:

```
module load nco
ncks -v TS,ICEFRAC *_01 jan.nc         (repeat for all 12 months of climo files)
```

35h_ii.

```
ncrcat jan.nc feb.nc (etc) jan-dec.nc   (concatenate all 12 mo's into 1 file)
```

35h_iii.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
                sst_cami_create_template.ncl
```

Rename the template file to something appropriate for your work, for example, sst_cami_create_PETM.ncl

- set new_cdf to sst_$atmres_$CASE_c<date>.nc   (name of new sst file)
- set cam_cdf to jan-dec.nc
- uncomment the setting for fv_cdf appropriate for your $atmres
- customize modtext to document origins of interpolated data

35h_iv.

```
ncl sst_cami_create_$CASE.ncl       (make sure ncl module is loaded)
```

35h_v.

```
./xmlchange SSTICE_DATA_FILENAME="sst_$atmres_$CASE_c<date>.nc"
```

35h_vi.
- customize user_nl_cice
  - set stream_fldfilename="sst_$atmres_$CASE_c<date>.nc"

35i. create interpolated cam initial conditions file

input:

```
<fully_coupled_run>.cam.i.yyyy-mm-dd-00000.nc   (cam initial condition file from end of
                                                  fully-coupled run)
/glade/p/cesm/cseg/inputdata/atm/cam/inic/fv/cami-mam3-0000-01-
01_0.23x0.31_L30_c110527.nc
                                                 (template file)
```

Note: other template files are available, depending on your resolution

output:

```
<fully_coupled_run>.cam.i.yyyy-mm-dd_$atmres.nc
```

35i_i.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
            interpic
```

Note: if using a machine other than cheyenne or executables require rebuild,
get source code and makefile to build new executable:

```
  > svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/
              interpic_src
  > cd interpic_src
  > make
  > cp interpic ../.
  > cd ..
```

35i_ii.

```
./interpic -v -t /glade/p/cesm/cseg/inputdata/atm/cam/inic/fv/cami-mam3-0000-01-
01_0.23x0.31_L30_c110527.nc <fully_coupled_run>.cam.i.yyyy-mm-dd-00000.nc
<fully_coupled_run>.cam.i.yyyy-mm-dd_$atmres.nc
```

35i_iii.
- customize user_nl_cam
  - set ncdata="<fully_coupled_run>.cam.i.yyyy-mm-dd_$atmres.nc"

35j.
perform steps 29 and 30 using 0.23x0.31 as $atmres
- customize user_nl_cam
  - set ncdata="bnd_topo_$CASE_$atmres_remap_sgh30.<date>.nc"

35k.
interpolate dst_* file from step 32 to 0.23x0.31
- customize user_nl_cam
  - set soil_erod with interpolated file

35l.
- customize user_nl_cam
  - set solar_data_file with file from step 31
  - set tropopause_climo_file with file from step 32
  - set tracer_cnst_datapath and tracer_cnst_file with file from step 32
  - set depvel_lnd_file with file from step 32
  - set ext_frc_specifier (SO2,so4_a1,so4_a2,num_a1,num_a2,bc_a1,pom_a1)
    with files from step 32
  - set srf_emis_specifier (SO2,so4_a1,so4_a2,num_a1,num_a2,bc_a1,SOAG,DMS,pom_a1)
    with files from step 32
  - customize other settings not addressed in this guide
    n2ovmr,ch4vmr,f11vmr,f12vmr
  - customize output settings
    fincl,nhtfrq,mfilt

35m.
perform steps 17 and 18 using 0.23x0.31 as $atmres
- modify user_nl_clm with new settings
  - set fsurdat='surfdata_$atmres_$CASE_c<date>.nc'

35n.
- customize output settings
  hist_fincl,hist_nhtfrq,hist_mfilt

35o.
- modify user_nl_cpl with customized settings from following steps in this guide
  - adjust orbital settings, if necessary

35p.
- modify user_nl_rtm with customized settings from following steps in this guide
  - set frivinp_rtm with file from step 24
  - customize output settings
    rtmhist_fincl,rtmhist_nhtfrq,rtmhist_mfilt

35q.
modify pe layout in env_mach_pes.xml (if necessary)
./cesm_setup
./<casename>.build

35r.
modify run settings in env_run.xml (length of run, etc)
modify run settings in <casename>.run (charge account, job queue, etc)
./<casename>.submit

36. **SLAB OCEAN for PETM**:
Setting up an atmosphere-only SOM (E case) model run at 2deg atmosphere:

Note: These instructions are specific to paleo (PETM) SOM runs and are recommended over the steps for the general SOM in code release

input: a spun-up, equilibriated and fully coupled CESM run

36a.
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
    slab_ocean_tools/pop_frc.csh

Edit the 'pop_frc.csh' script to specify the settings needed to create
a monthly-mean annual climatology ("MAC") file.

- Change CASE to the casename of the spun-up, equillibrated and fully-coupled CESM run
- Set BEGYR and ENDYR to the appropriate years for your climatology. 50yrs of data is typical.
- Set WKDIR to path where raw history files of CASE will reside and "MAC" file will be created
- Comment out read commands from the mass store (if appropriate)
- If there are coupler history files available, these can be used for the velocities and sea surface tilt terms. These are not required, however, and it is recommended to set CPLFILES to FALSE

36b. Stage pop monthly history files in $WKDIR

36c. Execute the .csh script
- module load nco
- tcsh pop_frc.csh > pop_frc.out &
  (takes maybe 5min for 50yrs of 320x384 data)
- creates $WKDIR/$CASE.pop.h.$BEGYR-$ENDYR.MAC.nc (ie, the "MAC" file)

36d.

```
svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/
            slab_ocean_tools/pop_frc_mlt.ncl
```

- set case to match what was used in pop_frc.csh
- set popmac to the "MAC" file just created (include path)
- set f3 to addfile an appropriate domain file for your run
  for example: /glade/p/cesmdata/cseg/inputdata/share/domains/
  domain.ocn.gx1v6.090206.nc

36e. execute the ncl script

- module load ncl
- ncl pop_frc_mlt.ncl
  (takes about 2min for 50yrs of 320x384 data)
- creates ./oceanmixed_ice.nc which has all the appropriate SOM forcing fields for your model run
- cp oceanmixed_ice.nc to /glade/p/cesm/cseg/inputdata/ocn/docn7/SOM/

Note: give it an appropriate name (ie pop_frc.gx1v6.<date>.nc)
The model run will point to this via the env_run.xml setting for DOCN_SOM_FILENAME

Note: if user has permission to copy this file to their inputdata directory (ie the directory noted in setting for DIN_LOC_ROOT in env_run.xml) then they may do so...if not, the user will have to modify the file:
<cesm_tag>/models/ocn/docn/bld/namelist_files/namelist_defaults_docn.xml
(this code change cannot be included as a SourceMod and must happen in the source code itself)

- set strm_domdir and strm_datdir to the path of the directory where the file exists

36f. Set up model run

```
./create_newcase -case /<case_directory>/<casename>
           -res f19_g16
           -mach cheyenne
           -compset EPETMC5CN
           -user_mods_dir  ../usermods_dirs/EPETMC5CN
```

Note: the arg "user_mods_dir" is used to pull in settings that match the PETM 2deg equillibrated and fully-coupled CESM run.

You can use this as a template for your period and adjust as necessary.

- cd to case directory
- ./cesm_setup
- ./<case>.build
- ./xmlchange DOCN_SOM_FILENAME=<som_forcing_file>
- modify run settings...
- ./<case>.submit

C. “Gotcha’s” (Typical runtime issues)

Gotcha's (Examples of potential runtime problems)

Gotcha1:

- Under extreme conditions of some paleo periods, it is not unusual for the model to dump warning statements that can potentially overwhelm the output logs. It might help to have some of these write statements commented out. Look for the 'write' statements in these files:

```
~models/atm/cam/src/dynamics/fv/fill_module.F90
~models/atm/cam/src/physics/cam/qneg3.F90
```

Any changes to source code would have to be included as SourceMods in your case directory and a rebuild would need to be performed:

```
./<casename>.build --clean
cp modified_source_code to <casedir>/SourceMods/src.xxx/.
./<casename>.build
```

Gotcha2:

- Running with an active ocean model (pop) under extreme conditions of some paleo periods can often lead to convergence problems in some of it's routines. Tweaking the pop time step can sometimes help. Restart from a point just before the failure (several days earlier, at least) and increase the setting for dt_count in user_nl_pop2 by 1 or 2. (Example: The highest we used for PETM was dt_count = 39 and we did not notice a performance hit).

Gotcha3:

- The Finite-Volume (fv) dynamics package can sometimes encounter instabilities - particularly at higer resolutions. Restarting the model just before failure (several days, at least) and tweaking the fv sub-cycling parameters can often get you past the problem. Some of the following suggestions may not run as efficiently as the default settings, so you may want to set them back after the model has moved past the trouble spot (perhaps several months of model time).

in user_nl_cam:

```
nsplit = 12
nspltrac = 6
nspltvrm = 2
```

can be replaced with:

```
nsplit = 12
nspltrac = 6
nspltvrm = 3
```

or

```
nsplit = 12
nspltrac = 12
nspltvrm = 6
```

Gotcha4:

-If none of the other recommended remedies are able to get you past a model crash, you might consider altering the CAM timestep. This requires a hybrid restart, which restarts the model clock for the atm and lnd, so it becomes a problem when stringing together the model history files. Also, the new timestep is likely to slow model performance, meaning you'll want to do another hybrid restart after a few months of model time when you're past the trouble spot. For these reasons, this should be considered a last resort. Hybrid restarts require a CAM initial file - this can be created by backing up to a restart just before the crash and running for one day with the CAM namelist setting, inithist='DAILY' enabled.