HELPFUL LINKS

- PaleoToolkit Instructions: https://github.com/CESM-Development/paleoToolkit/blob/master/cesm1_2/PaleoToolkit_Recipe_2020Jan1.pdf
- PaleoToolkit Resources: https://github.com/CESM-Development/paleoToolkit/tree/master/cesm1_2
- NCAR GitHub: https://github.com/NCAR/iCESM1.2
- UCAR Resources: https://wGitHub - NCAR/iCESM1.2: Isotope-enabled CESM1.2ww.cesm.ucar.edu/models/paleo/faq/
- CESM Distance Learning Course: https://www.meted.ucar.edu/training_module.php?id=1363#.XVtrkJNKjOR
- NCL Resources: https://www.ncl.ucar.edu/
- Notes: replace anything surrounded by < > with your specific naming conventions, anything in gray is a note, anything in black is code

---

TOPOGRAPHY / BATHYMETRY

- Log in
- module load nco ncl python ncarenv ncview
- cd /glade/work/<username>/
- mkdir <casename>
- cd <casename>
- mkdir topobath
- Find a global topography/bathymetry/paleogeography netcdf file and download it
- Log out and cd ~/Downloads → scp <folder/folder/filename> <username>@cheyenne.ucar.edu:/glade/work/<username>/<casename>/topobath
    - Log back in and find your downloaded netcdf → ncview or ncdump -h <netcdf> to find variable names
    - We want the variables to be named lat, lon, and topo
- ncrename -v z (original variable name),topo (new variable name) <netcdf>
- cp <netcdf> <netcdf2>
    - Copy and paste topobath file so we have 2 of the same - one for smoothing over and one for leaving alone (just incase)
- vi create0.5degree.v2.ncl (/glade/work/<username>/<casename>/topobath/)
    - fili = "<netcdf>"
    - filo = "topobath_0.5deg.nc"
    - ilat = fin->lat

    - ilon = fin->lon
    - topf = fin->topo
    - nlat = 360
    - nlon = 720
    - sl = -179.75 (for lon)
    - sl = -89.75 (for lat)
- ncl create0.5degree.v2.ncl
    - Now topobath is 0.5 deg, can also change to any other desired resolution

---

OCEAN

- svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/ocn/mk_ocn_grid/mk_grid_1x1_template.csh
- chmod +x everything (which often works to make sure you have permission to the files)
- In /glade/work/<username>/<casename>/topobath → cp topobath_0.5deg.nc /glade/work/<username>/<casename>/ocn/mk_ocn_grid/ for next step
- vi smooth_topo.ncl
    - ftopo1 = addfile("topobath_0.5deg.nc", "r")
    - ftopo2 = addfile("topobath_0.5deg.nc", "w")
    - fout_name = "smoothed_topobath_0.5deg.nc"
    - This script smooths sudden changes in depth in the ocean (ridges, coastlines, etc), but it can also be edited to smooth land; smoothing may alleviate potential errors
- ncl smooth_topo.ncl
- mv mk_grid_1x1_template.csh mk_grid_1x1_<casename>.csh
- vi mk_grid_1x1_<casename>.csh
    - CASE = <casename>
    - ITER = 1
    - topopath = /glade/work/<username>/<casename>/topobath
    - topofile = topobath_0.5deg.nc
    - vrtgridpath = /glade/work/<username>/<casename>/ocn/mk_ocn_grid
    - vrtgrid = gx1v6_vert_grid
    - lonnp = 0
    - latnp = 85
    - lonsp = 0
    - latsp = -85
        - You will change the location of the poles depending on your topobath file, but you can start with these numbers
    - dsig = 23.
        - Increasing this will increase the size of the poles
    - jcon = 11

        - Increasing this may help to move the poles inland
- ./mk_grid_1x1_<casename>.csh
- mv mk_SCRIPgrid_template.csh mk_SCRIPgrid_<casename>.csh
- vi mk_SCRIPgrid_<casename>.csh
    - popgriddir = /glade/work/<username>/<casename>/ocn/mk_ocn_grid
    - scripdir = /glade/work/<username>/<casename>/ocn/mk_ocn_grid
    - popgrid = grid.1.pop.da
    - kmtgrid = kmt.1.da
    - ocngridname = <casename>
    - ocnres = gx1<casename>
- ./mk_SCRIPgrid_<casename>.csh
- vi plot_all.sh
    - ocnres = gx1<casename>
    - date = <date> (see date from output file)
- ./plot_all.sh
- gv .ps files that are produced to see what the poles look like
- Change placement of the poles and repeat previous steps until poles are several grid cells from coasts
    - Same longitude for np and sp may help
    - May have to revisit this step if instabilities in model (moving poles, changing dsig, jcon, etc)
    - vi mk_grid_1x1_<casename>.csh → change anything, increase ITER by 1
    - ./mk_grid_1x1_<casename>.csh
    - vi mk_SCRIPgrid_<casename>.csh → change popgrid and kmtgrid to new ITER
    - ./mk_SCRIPgrid_<casename>.csh
    - vi plot_all.sh → change date (if new day)
    - ./plot_all.sh
    - Delete older iteration files once you're done
- mv mk_ocninput_template.csh mk_ocninput_<casename>.csh
- cp kmt.<ITER>.da /glade/work/<username>/<casename>/ocn/mk_ocninput/
- cp grid.<ITER>.pop.da /glade/work/<username>/<casename>/ocn/mk_ocninput/
- vi mk_ocninput_<casename>.csh
    - CASE to <casename>
    - GRID to grid.<ITER>.pop.da
    - KMT to kmt.<ITER>.da
    - You can choose to customize region_ids
        - You mostly change this if you have any marginal seas (optional)
    - You can choose to customize transport_contents

        - This is diagnostic so it won't change anything in the model but can offer information about transport in different regions if you want (would give you extra output files after running)
        - I left everything commented out (so not asking for this info)
- vi modregmsk_edit.f
    - Customize code if needed
    - I just added "double precision ulon(imt,jmt)" under the "double precision ulat(imt,jmt)" line
- ./mk_ocninput_<casename>.csh
- cp gx1v6_vert_grid /glade/work/<username>/<casename>/ocn/mk_ocninput/
- cp grid.<ITER>.plot.da /glade/work/<username>/<casename>/ocn/mk_ocninput/
- cp grid_bin2nc /glade/work/<username>/<casename>/ocn/mk_ocninput/
- ./grid_bin2nc
    - Lon dimension = 320
    - Lat dimension = 384
    - Name = region_mask_<casename>.nc
    - Depth profile = gx1v6_vert_grid
    - KMT = region.<casename>.be.ieeei4
    - Binary plot = grid.<ITER>.plot.da
    - Binary grid = grid.<ITER>.pop.da
    - Afterwards, output file ("region_mask_<casename>.nc") should have lat and lon as dimensions and several variables; the KMT variable should show defined ocean regions with ocean as 1 and land as 0
- vi cmpRegionMask2KMT.ncl
    - ifile1 = IPATH + "/kmt.<ITER>.da"
    - ifile2 = "region.<casename>.be.ieeei4"
- ncl cmpRegionMask2KMT.ncl
    - The last line should read "Region mask and KMT match exactly"

COUPLER

- In /glade/work/<username>/<casename> → mkdir cpl
- cp gx1<casename>_<date>.nc /glade/work/<username>/<casename>/cpl/
- cd /glade/p/cesm/cseg/inputdata/share/scripgrids/
- cp fv1.9x2.5_090205.nc /glade/work/<username>/<casename>/cpl
- cd /glade/work/<username>/<casename>/cpl
- ln -s /glade/work/juliacam/LOWP/2023/cpl/gen_cesm_maps.sh .
- ln -s /glade/work/juliacam/LOWP/2023/cpl/gen_ESMF_mapping_file/ .
- chmod +x everything
- cp gen_cesm_maps.sh gen_cesm_maps_<name>.sh

- /bin/csh
- setenv ESMFBIN_PATH /glade/u/apps/dav/opt/esmf/7.1.0r-ncdfio/intel/17.0.1/bin/binO/Linux.intel.64.mpiuni.default
- module purge
- module load gnu/8.3.0 esmf_libs/7.1.0r intel/18.0.5 intel/19.0.2 or whatever versions are needed when you try following module (try "module spider esmf-7.1.0r-defio-uni-O" to see how to load the module)
- module esmf-7.1.0r-defio-uni-O
- ./gen_cesm_maps_<name>.sh -fatm /glade/p/cesm/cseg/inputdata/share/scripgrids/fv1.9x2.5_090205.nc -natm fv19_25 -focn /glade/work/<username>/<casename>/cpl/gx1<casename>_<date>.nc -nocn gx1<casename> --nogridcheck
- svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/cpl/gen_domain
- ./gen_domain -m map_gx1<casename>_TO_fv19_25_aave.<date>.nc -o gx1<casename> -l 19_25 -p 2
- Some of these files may become outdated in time
- Important files to produce from the coupler to be used in simulation
    - domain.lnd.19_25_gx1<casename>.<date>.nc
    - domain.ocn.gx1<casename>.<date>.nc
    - map_fv19_25_TO_gx1<casename>_aave.<date>.nc
    - map_fv19_25_TO_gx1<casename>_blin.<date>.nc
    - map_fv19_25_TO_gx1<casename>_patc.<date>.nc
    - map_gx1<casename>_TO_fv19_25_aave.<date>.nc

LAND

- Need to make a global LSM netcdf that fits with chosen topobath file (SKIP this step if one is provided alongside topobath file)
    - LSM data and topo data used as inputs should have same resolution and have longitude ordered 0 to 360
    - Order topobath file from 0 to 360
        - In /glade/work/<username>/<casename>/ → mkdir lnd
        - cp topobath_0.5deg.nc /glade/work/<username>/<casename>/lnd/
        - cp topobath_0.5deg.nc topobath_0to360.nc
        - ncl
        - fn = "topobath_0to360.nc"
        - f1 = addfile(fn,"rw")
        - mytopo = f1->topo

        - mytopo = lonFlip(mytopo)
        - f1->topo = mytopo
        - exit
    - Convert topobath file to 2x2 deg resolution
        - cp /glade/work/juliacam/LOWP/2023/lnd/create_landmask_2x2_LOWP.ncl /glade/work/<username>/<casename>/lnd/
        - mv create_landmask_2x2_LOWP.ncl create_landmask_2x2_<casename>.ncl
        - vi create_landmask_2x2_<casename>.ncl
            - srcFileName = "topobath_0to360.nc"
                - May need to convert 0.5 deg file to 1deg before 2deg
            - temp = src_file->topo
            - filo = "topobath_2deg_0to360.nc"
            - fout->topo = temp_regrid
        - ncl create_landmask_2x2_<casename>.ncl
        - Make sure when converting your files to 2deg resolution that you use an interpolation method that does not smooth out integers, this will create problems in the raw surface datasets (e.g. use ESMF_regrid ncl commands with the interpolation method "neareststod")
    - Now the topobath file is ordered 0 to 360 and has 2x2 deg resolution
    - Now we need to use that to make a LSM file
        - cp /glade/work/juliacam/LOWP/2023/lnd/Mod_LOWP_veg.ncl /glade/work/<username>/<casename>/lnd/
        - mv Mod_LOWP_veg.ncl Mod_<casename>_veg.ncl
        - cp topobath_2deg_0to360.nc lsm_2deg_0to360.nc
        - ncrename -v topo,SUR lsm_2deg_0to360.nc
        - vi Mod_<casename>_veg.ncl
            - dir = "/glade/work/<username>/<casename>/lnd/"
            - case = "lsm_2deg_0to360.nc"
            - system("cp "+dir+case+" test.nc")
            - In = addfile(dir+"test.nc", "w")
            - pft = in->SUR
            - landmask = in->SUR
            - la = in->lat
            - lo = in->lon
            - do x = 0 , 179
            - do y = 0 , 89
            - Have all land = 1 at beginning, then fill in LSM types wherever, then have all ocean = 0 at end

        - Fill in ponds and lakes that are on land (check domain.lnd file) by ignoring elevation (ex below)
            - if (y.ge.87. .and. y.le.(91.) .and. x.ge.298. .and. x.le.305.) then
              in->SUR(y,x) = 10 ;testing lake
              end if
        - See /glade/work/juliacam/LOWP/2023/lnd/Mod_LOWP_veg.ncl for an example
    - ncl Mod_<casename>_veg.ncl
- svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/lnd/run_paleo_mkraw_cesm1_template.csh
- chmod +x everything
- mv run_paleo_mkraw_cesm1_template.csh run_paleo_mkraw_cesm1_<casename>.csh
- mv test.nc lsm_data.nc
- mv topobath_2deg_0to360.nc topo_2deg.nc
- vi run_paleo_mkraw_cesm1_<casename>.csh
    - CASE = <casename>
    - INPUT_LSM_DATA = lsm_data.nc
    - INPUT_TOP_DATA = topo_2deg.nc
- vi paleo_mkraw_cesm1_sed.F90
    - nlon = 180
    - nlat = 90
    - expected var name for topobath file is ‘topo’
    - expected var name for lsm file is ‘SUR’
    - topobath and lsm file names should be <80 characters
    - Start with default soil color (10) and modify later if desired
- cp /glade/p/cesm/palwg/paleo_setup/lnd/paleo_mkraw_cesm1/mksrf_zon_organic.10level.nc /glade/work/<username>/<casename>/lnd/
- In both paleo_mkraw_cesm1.F90 and run_paleo_mkraw_cesm1_<casename>.csh, the soil text file is hardcoded in “/glade/p/cesm/cseg/inputdata/lnd/clm2/rawdata/mksrf_soitex.10level.c010119.nc” but I kept running into an error that it couldn’t read the soil texture file so instead I copied it into my directory and gave myself permission to it and then changed the line in both those scripts to delete the other directory and just have the soil texture file = ‘mksrf_soitex.10level.c010119’ since I now had it in my own directory and then it worked and could read the file
- mv mkscripgrid_template.ncl mkscripgrid_<casename>.ncl
- vi mkscripgrid_<casename>.ncl

    - name = “<casename>”
    - fn1 = “mksrf_lanwat_<casename>.c<date>.nc”
- ncl mkscripgrid_<casename>.ncl
- mv mkmapdata_paleo.sh mkmapdata_<casename>.sh
- vi mkmapdata_paleo.sh
    - INGRID[nfile]=”./SCRIPgrid_<casename>_90x180.<date>.nc”
    - grids=(“2x2_<casename>”)
- mv regridbatch_paleo.sh regridbatch_<casename>.sh
- vi regridbatch_<casename>.sh
    - Set batch commands at top
    - resols=1.9x2.5
    - csmsrc=”/glade/p/cesm/palwg/cesm1_2_0”
- /bin/csh/
- setenv ESMFBIN_PATH /glade/u/apps/dav/opt/esmf/7.1.0r-ncdfio/intel/17.0.1/bin/binO/Linux.intel.64.mpiuni.default
- module load intel mpt esmf_libs nco ncl ncarcompilers ncarenv ncview
- qsub regridbatch_paleo.sh
    - qstat -u <username> to check the status
- mv mksurfdata_map.namelist.paleo mksurfdata_map.namelist.<casename>
- vi mksurfdata_map.namelist.<casename>
    - mapping_file = map_2x2_<casename>_to_1.9x2.5_nomask_aave_da_c<date>.nc
    - mksrf* files = xxx_<casename>.c<date>.nc
    - out_res = 1.9x2.5
    - casename = <casename>
    - date = <date>
    - This file has a ton of lines to modify, but it says where to put your mapping file vs your mksrf files, fix the out_res, casename, and date for output files
- ./mksurfdata_map < mksurfdata_map.namelist.<casename>
- I ran into an error that a soil color of 10 was not currently supported (though it was the default) so I went into “paleo_mkraw_cesm1_sed.F90” and changed the soil color to 20 (“soil_color(j,i) = 20._r8”) then reran “run_paleo_mkraw_cesm1_<casename>.csh” to create new mksrf files (today was a new date so it didn’t replace the old ones) then I went into mksurfdata_map.namelist.<casename> and changed “mksrf_fsoicol” to the new mksrf_soicol file with today’s date but left all the others the same then reran mksurfdata_map < mksurfdata_map.namelist.<casename> and it worked
- Make sure your final surfdata*.nc files make sense (especially PFT variables → check out PCT_PFT in 3d variables)

RUNOFF

- Use a topobath file that is 0.5x0.5 deg and ordered 0 to 360 (lon) with variable name topo and dimension names lat and lon
- In /glade/work/<username>/<casename>/ → mkdir rof
- svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/rof/rdirc_template.csh
- mv rdirc_template.csh rdirc_<casename>.csh
- vi rdirc_<casename>.csh
    - INFILE = topobath_0.5deg_0to360.nc
    - CASE = <casename>
- vi topo2rdirc_sed.F90
    - nlon = 720
    - nlat = 360
- ./rdirc_<casename>.csh
    - There may be a few pauses, just hit "enter" at each prompt, AKA no need to actually write anything at the prompts
- vi plotrdirc.csh
    - IFILE = topobath_0.5deg_0to360.nc
    - NLAT = 360
    - NLON = 720
    - RESOLN = 0.5x0.5
    - CASE = <casename>
- Modify plot_rdirc.ncl if you want to zoom in on geographical area of interest (modify lats and lons under section labeled "Zoom in on data") or leave all commented out if you want a global plot
- ./plotrdirc.csh
    - This step is optional, it's to see what the runoff map looks like so far - it's a good idea to check it and make sure nothing looks unexpected, feel free to zoom in on any areas and rerun this script to make a new map
- vi check_inf_loop.F90
    - nlat = 360
    - nlon = 720

    - filei = ‘rdirc.0.5x0.5.<casename>’
- make check_inf_loop
- ./check_inf_loop this is optional but it checks that there are no infinite loops, if it pauses at a prompt again just hit “enter” like before
- vi rtm_ncdf.pro
    - rtmfile1 = ‘rdirc.0.5x0.5.<casename>’
    - outfile = ‘rdirc.0.5x0.5.<casename>.nc’
    - resnum=1
        - resnum=2 for 1x1 deg, resnum=1 for 0.5x0.5 deg
    - Set history attribute with username and date
    - Set source attribute with casename
- module load idl
- idlde this opens IDL, once it loads (will be slow), change the “Current Directory” to whatever directory you’ve been working in (that has rtm_ncdf)
- .rn rtm_ncdf this compiles the RTM module
- rtm this executes the code
- Exit IDL
- mv runoff_map.1x1.template.nml runoff_map.nml needs to be called runoff_map.nml to work in next step
- cp -r /glade/work/mlevy/codes/cime /glade/work/<username>/<casename>/rof
- cp /glade/work/<username>/<casename>/ocn/mk_ocn_grid/gx1<casename>_<date>.nc /glade/work/<username>/<casename>/rof/cime/tools/mapping/gen_mapping_files/runoff_to_ocn
- cp runoff_map.nml /glade/work/<username>/<casename>/rof/cime/tools/mapping/gen_mapping_files/runoff_to_ocn
- cp rdirc.0.5x0.5.<casename> /glade/work/<username>/<casename>/rof/cime/tools/mapping/gen_mapping_files/runoff_to_ocn
- Need most updated runoff_map and SCRIP mapping file for next step to work
- vi runoff_map.nml
    - file_roff = ‘./rdirc.0.5x0.5.<casename>’
    - file_ocn = ‘./gx1<casename>_<date>.nc’

    - Replace <casename> with <casename>, <ocnres> with gx1<casename>, <date> with the date, and resolution with 05x05

- ./runoff_map < runoff_map.nml
- cp /glade/p/cesmdata/cseg/inputdata/share/scripgrids/r05_nomask_070925.nc /glade/work/<username>/<casename>/rof/cime/tools/mapping/gen_mapping_files/runoff_to_ocn
- module purge
- module load intel nco esmf_libs
- ./create_ESMF_map.sh -fsrc r05_nomask_070925.nc -nsrc r05_nomask -fdst ./gx1<casename>_<date>.nc -ndst gx1<casename> -map aave
- cp any files you created in this directory that are important (nnsm, aave mapping files) to your regular rof directory to keep it all in the same place
- Important files to produce from the coupler to be used in simulation
    - map_r05_nomask_TO_gx1v6<casename>_aave.<date>.nc
    - map_r05x05<casename>_to_gx1<casename>_nnsm_e1000r300_<date>.nc

---

ATMOSPHERE

- I used the 0.5 degree topobath file ordered 0 to 360 longitude that I had in my lnd directory, variable should be 'topo'
- In /glade/work/<username>/<casename>/ → mkdir atm
- svn export https://github.com/CESM-Development/paleoToolkit/trunk/cesm1_2/atm/mk_10min_definesurf_input_paleo.ncl
- vi mk_10min_definesurf_input_paleo.ncl
    - cases = (/"<casename>"/)
    - path = "/glade/work/<username>/<casename>/atm/"
    - topoinput = "topobath_0to360.nc"
    - ncl script should already point to USGS file
- ncl mk_10min_definesurf_input_paleo.ncl
- cd /glade/p/cesm/palwg/paleo_setup/atm/definesurf/public
- cp USGS-gtopo30_1.9x2.5_remap_c050602.nc /glade/work/<username>/<casename>/atm
- cp landm_coslat.nc /glade/work/<username>/<casename>/atm
- ./definesurf -remap -r -t <casename>_10min_topo_4input2definesurf.<date>.nc -g USGS-gtopo30_1.9x2.5_remap_c050602.nc -l landm_coslat.nc bnd_topo_<casename>_1.9x2.5_remap.<date>.nc (this produces bnd_topo_<casename>_1.9x2.5_remap.<date>.nc)

    - Another way to get this file: modify mk_cesm1_cami_bnd_topo_paleo.ncl with 1 deg 0 to 360 topobath file, USGS file, and 1.9x2.5 <casename> surfdata file and then run it by ncl
    - Even though these were 2 different methods, they both seem to have made the same file! I am using "bnd_topo_<casename>_1.9x2.5_remap.<date>.nc" file moving forward
- vi add_SGH30_paleo.ncl
    - cases = (/"<casename>"/)
    - ifile1 = "bnd_topo_<casename>_1.9x2.5_remap.<date>.nc"
    - ofile = "bnd_topo_<casename>_1.9x2.5_remap_sgh30.<date>.nc"
- cd /glade/p/cesmdata/cseg/inputdata/atm/cam/solar/
- cp SOLAR_SPECTRAL_Lean_1610-2008_annual_c090324.nc /glade/work/<username>/<casename>/atm
- vi sol_constant_paleo.ncl
    - infil = "SOLAR_SPECTRAL_Lean_1610-2008_annual_c090324.nc"
    - outfil = "./solar_scon_<casename>.<date>.nc"
    - So_adj = 1358.38
    - Comment out the last 2 lines above "end" – they should just be about comments themselves
    - Chose solar constant based on Gough (1981): $[1+0.4(1-t/4.7)]^{-1}$
        - t is time in billions of years (4.7-0.025=4.675 for 25 Ma)
        - This yields the percent of the modern-day solar constant that you should use (.997877 x 1361.27 W/m^2 = 1358.38 W/m^2 for 25 Ma)
            - Using 1361.27 W/m^2 because that is 1850 CAM4 default
- ncl sol_constant_paleo.ncl
- Ok now you just need to finalize aerosols which is done with an initial cam file you get from a 1 year run
- So skip to initial 1 year run instructions (no isotopes), then return here after the 1 year run to finalize your aerosols!
- mv modify_aerosol_input_template.ncl modify_aerosol_input_<casename>.ncl
- vi modify_aerosol_input_<casename>.ncl
    - Pfil = "/glade/work/<username>/<casename>/cpl/domain.lnd.19_25_gx1<casename>.<date>.nc"
        - Need to fill this in in 2 places
    - Fp = "/glade/work/<username>/<casename>/atm/<casename1>.cam.i.0002-01-01-00000.nc"
    - top_in = "/glade/work/<username>/<casename>/atm/topobath_0.5deg.nc"
    - output_stem = "/glade/work/<username>/<casename>/atm/"

- ncl modify_aerosol_input_<casename>.ncl

INITIAL RUN (1 year - no isotopes)

- module load nco ncl ncarenv ncview ncarcompilers mpt intel netcdf
- In 'glade/work/<username>' : svn export /glade/work/juliacam/cesm1_2_2_1/
- mkdir case
- cd /glade/work/<username>/cesm1_2_2_1/scripts/
- ./create_newcase -case /glade/work/<username>/case/<casename1> -res f19_g16 -compset B1850C5CN -mach cheyenne
    - Choose appropriate resolution and compset for your case, machine may be different supercomputer (i.e. derecho)
- cd /glade/work/<username>/case/<casename1>/
- ./xmlchange CCSM_CO2_PPMV="280.0"
    - This is a pre-industrial CO2 example, but choose what's appropriate for your case
- Insert domain filenames
    - ./xmlchange ATM_DOMAIN_FILE="domain.lnd.19_25_gx1<casename>.<date>.nc"
    - ./xmlchange ATM_DOMAIN_PATH="/glade/work/<username>/<casename>/cpl"
    - ./xmlchange LND_DOMAIN_FILE="domain.lnd.19_25_gx1<casename>.<date>.nc"
    - ./xmlchange LND_DOMAIN_PATH="/glade/work/<username>/<casename>/cpl"
    - ./xmlchange ICE_DOMAIN_FILE="domain.ocn.gx1<casename>.<date>.nc"
    - ./xmlchange ICE_DOMAIN_PATH="/glade/work/<username>/<casename>/cpl"
    - ./xmlchange OCN_DOMAIN_FILE="domain.ocn.gx1<casename>.<date>.nc"
    - ./xmlchange OCN_DOMAIN_PATH="/glade/work/<username>/<casename>/cpl"
- Insert mapping filenames
    - ./xmlchange ATM2OCN_FMAPNAME="/glade/work/<username>/<casename>/cpl/map_fv19_25_TO_gx1<casename>_aave.<date>.nc"
    - ./xmlchange ATM2OCN_SMAPNAME="/glade/work/<username>/<casename>/cpl/map_fv19_25_TO_gx1<casename>_blin.<date>.nc"
    - ./xmlchange ATM2OCN_VMAPNAME="/glade/work/<username>/<casename>/cpl/map_fv19_25_TO_gx1<casename>_patc.<date>.nc"
    - ./xmlchange OCN2ATM_FMAPNAME="/glade/work/<username>/<casename>/cpl/map_gx1<casename>_TO_fv19_25_aave.<date>.nc"

    - ./xmlchange OCN2ATM_SMAPNAME="/glade/work/<username>/<casename>/cpl/map_gx1<casename>_TO_fv19_25_aave.<date>.nc"
    - ./xmlchange ROF2LND_FMAPNAME="lnd/clm2/mappingdata/maps/1.9x2.5/map_0.5x0.5_nomask_to_1.9x2.5_nomask_aave_da_c120709.nc"
    - ./xmlchange ROF2LND_SMAPNAME="lnd/clm2/mappingdata/maps/1.9x2.5/map_0.5x0.5_nomask_to_1.9x2.5_nomask_aave_da_c120709.nc"
    - ./xmlchange LND2ROF_FMAPNAME="lnd/clm2/mappingdata/maps/1.9x2.5/map_1.9x2.5_nomask_to_0.5x0.5_nomask_aave_da_c120522.nc"
    - ./xmlchange ROF2OCN_FMAPNAME="/glade/work/<username>/<casename>/rof/map_r05_nomask_TO_gx1v6<casename>_aave.<date>.nc"
    - ./xmlchange ROF2OCN_RMAPNAME="/glade/work/<username>/<casename>/rof/map_r05x05<casename>_to_gx1<casename>_nnsm_e1000r300_<date>.nc"
- Insert chosen PE layout
    - ./xmlchange NTASKS_ATM=468,NTHRDS_ATM=2,ROOTPE_ATM=0
    - ./xmlchange NTASKS_LND=144,NTHRDS_LND=1,ROOTPE_LND=0
    - ./xmlchange NTASKS_ICE=324,NTHRDS_ICE=1,ROOTPE_ICE=144
    - ./xmlchange NTASKS_OCN=108,NTHRDS_OCN=2,ROOTPE_OCN=468
    - ./xmlchange NTASKS_CPL=468,NTHRDS_CPL=2,ROOTPE_CPL=0
    - ./xmlchange NTASKS_ROF=144,NTHRDS_ROF=1,ROOTPE_ROF=0
    - This may change depending on your case
- ./cesm_setup
- vi env_run.xml
    - type=startup, startdate=0001-01-01, nyears=1, contine_run=false, etc
- vi user_nl_* files
    - Fill out with proper or standard initial condition files
    - See /glade/work/juliacam/case/LOWP3.1xC.001 for an example
- ./<casename1>.build
    - <casename1>.clean_build first if you're rebuilding after a failed attempt
- vi <casename1>.run
    - Set #PBS -A to charge proper account
    - Set walltime to proper amount (2 hours should be plenty for 1 year)
- ./<casename1>.submit
    - qstat -u <username> to check status (queued, running, or ended)
    - gladequota to check storage space

INITIAL RUN (5 days - isotopes)

- module load nco ncl ncarenv ncview ncarcompilers mpt intel netcdf
- In 'glade/work/<username>' : svn export /glade/work/juliacam/iCESM1_2_0_1_geotrace_n03/
- cd /glade/work/<username>/iCESM1_2_0_1_geotrace_n03/scripts/
- ./create_newcase -case /glade/work/<username>/case/<casename2> -res f19_g16 -compset B1850C5CN -mach cheyenne
- Choose same resolution and compset as above 1 year run
- Same xmlchanges as above but add…
- ./xmlchange -file env_build.xml -id CAM_CONFIG_OPTS -val "-water_tracer h2o_h216o_hdo_h218o" -append
- ./xmlchange OCN_TRACER_MODULES="iage wiso"
- ./cesm_setup
- vi env_run.xml
  - type=startup, startdate=0001-01-01, ndays=5, contine_run=false, etc
- vi user_nl_* files
  - Fill out with proper or standard initial condition files
  - See /glade/work/juliacam/case/LOWP1.WISO.5day.003 as an example
- ./<casename2>.build
  - <casename2>.clean_build first if you're rebuilding after failed attempt
- vi <casename2>.run
  - Set #PBS -A to charge proper account
  - Set walltime to proper amount (30 minutes should be plenty for 5 days)
- ./<casename2>.submit
  - qstat -u <username> to check status (queued, running, or ended)
  - gladequota to check storage space
- Notes
  - You will need some initial condition files, certain ones can be taken from NCAR
  - Paleotoolkit initial run - https://svn-ccsm-models.cgd.ucar.edu/cesm1/exp_tags/pcesm_cesm1_2_2_tags/cesm-dt2.0_cesm1_2_2_1/usermods_dirs/BPETMC5CN/
- After this run, there are some changes to make before a long, isotope-enabled run…
- Use interpinic tool
  - cp /glade/work/juliacam/LOWP/2023/lnd/LOWP1/interpinic /glade/work/<username>/<casename>/lnd/
  - cp /glade/scratch/<username>/<casename2>/run/<casename2>.clm2.r.0001-01-06-00000.nc /glade/work/<username>/<casename>/lnd/

- cp /glade/work/juliacam/LOWP/2023/lnd/LOWP1/b.e11.B1850C5CN.f19_g16.008.clm2.r.1001-01-01-00000.nc /glade/work/<username>/<casename>/lnd/
- ./interpinic -i b.e11.B1850C5CN.f19_g16.008.clm2.r.1001-01-01-00000.nc -o <casename2>.clm2.r.0001-01-06-00000.nc
- mv <casename2>.clm2.r.0001-01-06-00000.nc b.e11.B1850C5CN.f19_g16.008.clm2.r.1001.IP.<casename2>.nc
- cp b.e11.B1850C5CN.f19_g16.008.clm2.r.1001.IP.<casename2>.nc b.e11.B1850C5CN.f19_g16.008.clm2.r.1001.IP.<casename2>.WISO.nc
- cp /glade/work/juliacam/LOWP/2023/lnd/LOWP1/iclm4_IC_create.ncl /glade/work/<username>/<casename>/lnd/
- vi iclm4_IC_create.ncl
  - cdir = "/glade/work/<username>/<casename>/lnd/"
  - cfil = " b.e11.B1850C5CN.f19_g16.008.clm2.r.1001.IP.<casename2>.nc"
  - nfil = " b.e11.B1850C5CN.f19_g16.008.clm2.r.1001.IP.<casename2>.WISO.nc"
  - tfil = "/glade/work/<username>/<casename>/lnd/iso_clm4_IC.nc"
  - out = addfile("/glade/work/<username>/<casename>/lnd/iso_clm4_IC.nc","c")
- ncl iclm4_IC_create.ncl
- cp /glade/work/<username>/<casename>/lnd/b.e11.B1850C5CN.f19_g16.008.clm2.r.1001.IP.<casename2>.WISO.nc /glade/scratch/<username>/<casename2>/run/

- Set iCICE initial condition
  - cp /glade/work/juliacam/LOWP/2023/ocn/iCICE_IC/iCICE_IC_create.ncl /glade/work/<username>/<casename>/ocn/
  - cp /glade/scratch/<username>/<casename2>/run/<casename2>.cice.r.0001-01-06-00000.nc /glade/work/<username>/<casename>/ocn/
  - cp <casename2>.cice.r.0001-01-06-00000.nc <casename2>.cice.r.0001-01-06-00000.WISO.nc
  - vi iCICE_IC_create.ncl
    - dir = "/glade/work/<username>/<casename>/ocn/"
    - fname1 = "<casename2>.cice.r.0001-01-06-00000.nc"
    - fname2 = "<casename2>.cice.r.0001-01-06-00000.WISO.nc"
  - ncl iCICE_IC_create.ncl
  - cp <casename2>.cice.r.0001-01-06-00000.WISO.nc /glade/scratch/<username>/<casename2>/run/
- Creating WISO initial file

- cp /glade/work/juliacam/LOWP/2023/ocn/wiso_init_file/iPOP_R18O_HybFile_smth.ncl /glade/work/<username>/<casename>/ocn/
- cp /glade/work/juliacam/LOWP/2023/ocn/wiso_init_file/b.e12.B1850C5CN.f19_g16.iPI.01.pop.h.0400-0499.climo.nc /glade/work/<username>/<casename>/ocn/
- Also copy and paste a pop.h file from your 1 year no isotope run (casename1) and your 5 day isotope run (casename2)
- vi iPOP_R18O_HybFile_smth.ncl
  - case0name = "b.e12.B1850C5CN.f19_g16.iPI.01.pop.h.0400-0499.climo.nc" PI case
  - DIR0 = "/glade/work/<username>/<casename>/ocn/" have everything in this directory
  - case1name = "<casename1>.pop.h.0001-01.nc" 1 yr no isotope run
  - case2name = "<casename2>.pop.h.0001-01.nc" 5 day isotope run
  - system("rm -f "+"Init.<casename>_SMTH3_WISO.pop.h.climo.nc")
    - It'll want this to exist already, you can use my file or blank kmt
  - system("cp "+DIR0+"Init_WISO.pop.h.climo.nc "+DIR0+"Init.<casename>_SMTH3_WISO.new.pop.h.climo.nc")
    - It'll want this first one to exist already, you can use my file or blank kmt
  - addA = addfile(DIR0+"Init.<casename>_SMTH3_WISO.new.pop.h.climo.nc", "w")

- Now you're ready for a long run! Here are some notes for that …
  - Your first submission for any new case should have continue_run=false, and then change it to true after the first run if you're continuing to spin up that case
  - New clm2.r file from interpinic step is now used as finidat file in user_nl_clm
  - New cice.r file from iCICE step is now used as ice_ic file in user_nl_cice
  - New WISO file is now used as init_wiso_init_file in user_nl_pop2 (run it for first run, then when continue_run=true, comment out init_wiso_init_file, init_wiso_init_file_fmt, and init_wiso_option)
  - Also comment out init_ts_file and init_ts_file_fmt and change init_ts_option to 'mean' in user_nl_pop2 once continue_run=true
  - After a successful run with isotopes, you may want to create a new branch case and use that first successful submission as reference case or just continue as startup case
  - Look at timing directory to determine how many years can be run in 12 hour wall-clock limit, give yourself some cushion incase it takes longer

- Once you have a successful long run (~20 years), you can choose to resubmit as many times as you want
- Check gladequota often so you don’t run out of space, delete old files in scratch directory you no longer need or average all then delete monthly files, check TOA balance as you run until at equilibrium