**Building and running a FATES simulation (as of Feb 20, 2018)**

*Open Cygwin and type:*

xwin -multiwindow -nolisten tcp

To update

git submodule update --init --recursive

*Then open xterm from xwin.*

--------------------------------------------------------------------------------------------------------------------

*Log into LANL HPC in xterm:*

ssh -Y username[@wtrw.lanl.gov](mailto:username@wtrw.lanl.gov)

*Enter cryptocard password and then ssh into the wolf supercomputer (wf). Fe1 stands for the first front end node of the wolf machine. You may have to change this depending on which server you would like to access:*

ssh ba-fe1.lanl.gov

git clone [git@github.com:E3SM-Project/E3SM.git](about:blank) ACME

cd ACME

git submodule update --init --recursive

------------------

*Go to the script directory.*

cd $ACME\_ROOT/cime/scripts

*Check if the compset is valid. Note that there’s a period before the backslash.*

./query\_config --compsets clm

*Name the case. You can call the case whatever you want:*

set ACME\_CASE = WhateverYouWant

--------------------------------------------------------------------------------------------------------------------*Here are three examples of creating a new case. The command create\_newcase creates a case with the name you assigned (WhateverYouWant above) and assigns compset, resolution, compiler, etc…*

*Here we create a case to run a one degree by one degree simulation. Note that there’s a period before the backslash.*

./create\_newcase -case $CASE\_ROOT/$ACME\_CASE -compset ICLM45ED -res 1x1\_brazil -mach badger -mpilib mpi-serial -compiler intel

./create\_newcase -case $CASE\_ROOT/$ACME\_CASE -compset ICLM45ED -res 1x1\_brazil -mach wolf -mpilib mpi-serial -project w17\_treemortality -compiler intel

*Here’s the setup to run a global simulation:*

./create\_newcase -case $CASE\_ROOT/$ACME\_CASE -compset ICLM45ED -res f45\_g37 -mach wolf -compiler intel

--------------------------------------------------------------------------------------------------------------------

*Now go to the directory where the case is to specify the location and climate driver information.*

cd $CASE\_ROOT/$ACME\_CASE

*Here’s an example that you will need to customize to your specific simulation requirements:*

*Here is how to specify the input data.*

./xmlchange  DIN\_LOC\_ROOT=/usr/projects/cesm/input\_data

./xmlchange DIN\_LOC\_ROOT\_CLMFORC=/usr/projects/cesm/input\_data/atm/datm7

For BCI:

./xmlchange  DIN\_LOC\_ROOT=/usr/projects/cesm/input\_data

./xmlchange DIN\_LOC\_ROOT\_CLMFORC=/usr/projects/cesm/input\_data/atm/datm7

*Here is how you would change the domain. Note Word forces a hard return after the ./xmlchange command that doesn’t actually exist. Note also that two xml files need to be changed to change the domain.*

./xmlchange LND\_DOMAIN\_FILE=domain.lnd.1x1\_Colorado10\_FATES\_1x1\_Colorado10\_FATES.161222.nc

./xmlchange

ATM\_DOMAIN\_FILE=domain.lnd.1x1\_Colorado10\_FATES\_1x1\_Colorado10\_FATES.161222.nc

./xmlchange LND\_DOMAIN\_FILE=domain\_Delaware\_PT\_c180619.nc

./xmlchange ATM\_DOMAIN\_FILE=domain\_Delaware\_PT\_c180619.nc-

./xmlchange LND\_DOMAIN\_FILE=domain.lnd.1x1pt-brazil\_navy.090715.nc

./xmlchange ATM\_DOMAIN\_FILE=domain.lnd.1x1pt-brazil\_navy.090715.nc

*Here is an example of how set the domain to Yellowstone national park.*

./xmlchange LND\_DOMAIN\_FILE=domain.lnd.1x1\_YellowstoneNP\_1x1\_YellowstoneNP.171221.nc

./xmlchange ATM\_DOMAIN\_FILE=domain.lnd.1x1\_YellowstoneNP\_1x1\_YellowstoneNP.171221.nc

--------------------------------------------------------------------------------------------------------------------

*Now we setup the case and build it:*

./case.setup

./case.build

*Before rebuilding a case consider clearing out the previous attempts:*

./case.build --clean-all

--------------------------------------------------------------

#@Create a parameter file for FATES

cd $ACME\_ROOT/components/clm/src/external\_models/fates/parameter\_files

ncgen -o {parameter\_file\_name.nc} fates\_params\_coastal\_veg.cdl

--------------------------------------------------------------------------------------------------------------------

*Change the parameter input and surface data, type the following in xterm and create a new Fortran 90 file:*

nedit user\_nl\_clm

*Then paste the following into the newly created file*

! this line changes the surface file

fsurdat = '/usr/projects/cesm/input\_data/lnd/clm2/surfdata\_map/surfdata\_1x1\_YellowstoneNP\_16pfts\_simyr2000\_c171221.nc'

fsurdat = '/usr/projects/cesm/input\_data/lnd/clm2/surfdata\_map/surfdata\_Delaware\_PT\_c180619.nc'

! user parameter files

fates\_paramfile=’ /turquoise/usr/projects/veg/cxu/ACME/components/clm/src/external\_models/fates/parameter\_file/{parameter\_file\_name.nc}

fates\_paramfile = '/turquoise/usr/projects/veg/liangwei/paramfiles/fates\_params.c171120\_13pfts\_Wei\_Dec0817\_2xstorage\_2Mpa.nc'

! start from observed inventory?

use\_fates\_inventory\_init = .true.

fates\_inventory\_ctrl\_filename = '/turquoise/usr/projects/veg/liangwei/restart\_files/Restart\_Nov2017/yellowstone\_Jan18.txt'

! user output frequency and paramters

hist\_mfilt = 12,365

hist\_nhtfrq = 0,-24

hist\_fincl1='M1\_SCPF','M2\_SCPF','M3\_SCPF','M4\_SCPF','Md13c\_SCPF','Md13cRate\_SCPF','C13disc\_SCPF'

hist\_fincl2='M1\_SCPF','M2\_SCPF','M3\_SCPF','M4\_SCPF','Md13c\_SCPF','Md13cRate\_SCPF','C13disc\_SCPF'

--------------------------------------------------------------------------------------------------------------------*Change wall time if needed. In general if you’re running on a single node, budgeting 15 minutes per year of simulation time should give you an adequate buffer:*

nedit  env\_batch.xml

*Then change the following line:*

<queue default="true" walltimemax="16:00:00">standard</queue>

Change the run time in $CASE\_ROOT/$ACME\_CASE

nedit env\_run.xml

change STOP\_OPTION and STOP\_N to the desired run time.

--------------------------------------------------------------------------------------------------------------------*There are a number of ways to submit a job depending on whether you would like to run the job on a single node or on a cluster of nodes. In general, the computer gnomes frown upon running anything longer than a few minutes on the front end node.*

*To Submit the job if running on multiple nodes:*

./case.submit

*To submit the job if you’re running a short simulation on the single front-end node:*

./case.submit --no-batch

*Here is the preferred method if you’re running a longer simulation. You are running on a single wolf back end node. You use QOS interactive to ask for a single node:*

salloc -N 1 -t 240 --qos=interactive

*This requests 1 node, for 60 minutes, with interactive QOS. Then you submit as you would if you were running on the front-end:*

./case.submit --no-batch

--------------------------------------------------------------------------------------------------------------------*To kill/cancel a job:*

scancel *job-id#*

--------------------------------------------------------------------------------------------------------------------*To check the status of your simulation:*

*Check machine usage:*

sinfo

*To see were your request is in the line up:*

squeue –u dgoodsman

*To check the priority of your request:*

sprio –l

*Here are more details on scheduling policies if you are interested:*

<https://hpc.lanl.gov/scheduling_policies>

--------------------------------------------------------------------------------------------------------------------

*To check whether the simulation is finished type:*

cat CaseStatus

--------------------------------------------------------------------------------------------------------------------

*To check results (before the simulation is finished) type:*

cd $RUN\_ROOT/$ACME\_CASE/run

ls –lrt

ncrcat $ACME\_case….h0\*.nc all.nc

ncview all.nc

*This opens a GUI that you can use to preview the simulation results.*

Model debugging

##rebuild the case in debug mode

cd $CASE\_ROOT/$ACME\_CASE/

./xmlchange DEBUG=TRUE

./xmlchange CHECK\_TIMING=FALSE

./case.build --clean-all

./case.build

##run the model under totalview for bebugging

module load totalview

#rebuild the case

cd $CASE\_ROOT/$ACME\_CASE

./xmlchange DEBUG=TRUE

./case.build –clean-all

./case.build

#run the model with totalview  
cd $RUN\_ROOT/$ACME\_CASE/run

#set the running envrionment

source $CASE\_ROOT/$ACME\_CASE/.env\_match\_specific.csh

totalview ../bld/e3sm.exe

Link directory for input data

<http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/x1045.html>

cd $ACME\_ROOT/cime/scripts

./link\_dirtree –h

|  |
| --- |
| #!/bin/csh -f |
|  |  |
|  | if ($#argv != 2) then |
|  | cat << EOF1 |
|  |  |
|  | NAME |
|  |  |
|  | link\_dirtree - creates a virtual copy of one dir tree to another. the |
|  | directory structure is duplicated and the files are linked. |
|  |  |
|  | SYNOPSIS |
|  |  |
|  | link\_dirtree orig\_dir new\_dir |
|  |  |
|  | EXAMPLES |
|  |  |
|  | link\_dirtree /fs/cgd/csm/inputdata /ptmp/$LOGNAME/my\_inputdata |
|  |  |
|  | EOF1 |
|  | exit -1; |
|  | endif |
|  |  |
|  |  |
|  | #set orig\_dir = /fs/cgd/csm/inputdata |
|  | #set new\_dir = /ptmp/$LOGNAME/inputdata |
|  |  |
|  | set orig\_dir = $argv[1] |
|  | set new\_dir = $argv[2] |
|  |  |
|  | set cdir = `pwd` |
|  |  |
|  | if (! -e ${orig\_dir}) then |
|  | echo "orig\_dir does not exist : ${orig\_dir}" |
|  | exit -1 |
|  | endif |
|  | cd ${orig\_dir} |
|  | set odir = `pwd` |
|  |  |
|  | cd ${cdir} |
|  | if (! -e ${new\_dir}) mkdir -p ${new\_dir} |
|  | if (! -e ${new\_dir}) then |
|  | echo "exit from ${new\_dir}" |
|  | exit -2 |
|  | endif |
|  | cd ${new\_dir} |
|  | set ndir = `pwd` |
|  |  |
|  | echo " " |
|  | echo "running link\_dirtree (this could take a minute)" |
|  | echo "orig\_dir = $odir" |
|  | echo "new\_dir = $ndir" |
|  |  |
|  | cd ${odir} |
|  |  |
|  | # tcraig, this doesn't work on some platforms |
|  | #find . -type d -exec mkdir -p "${new\_dir}/{}" \; |
|  | #find . -type f -exec ln -s -f "${orig\_dir}/{}" "${new\_dir}/{}" \; |
|  |  |
|  | # tcraig, this doesn't work on bangkok |
|  | find . -type d | xargs -L 1 -I {} mkdir -p ${ndir}/\{\} |
|  | find . -type f | xargs -L 1 -I {} ln -s -f ${odir}/\{\} ${ndir}/\{\} |
|  |  |
|  | cd ${cdir} |
|  |  |
|  | exit 0 |