

# Running CLM5.0: CHESS Practical Session

Danica Lombardozzi



# Outline

### CESM at a glance

- 1) The CESM framework
- 2) Finding information about CLM & CESM
- 3) Overview of CLM (and CESM) directory structure

### Basic workflow

- 1) Create a new case
- 2) Invoke case\_setup
- 3) Build the executable
- 4) Run and output data

### Changing configuration options

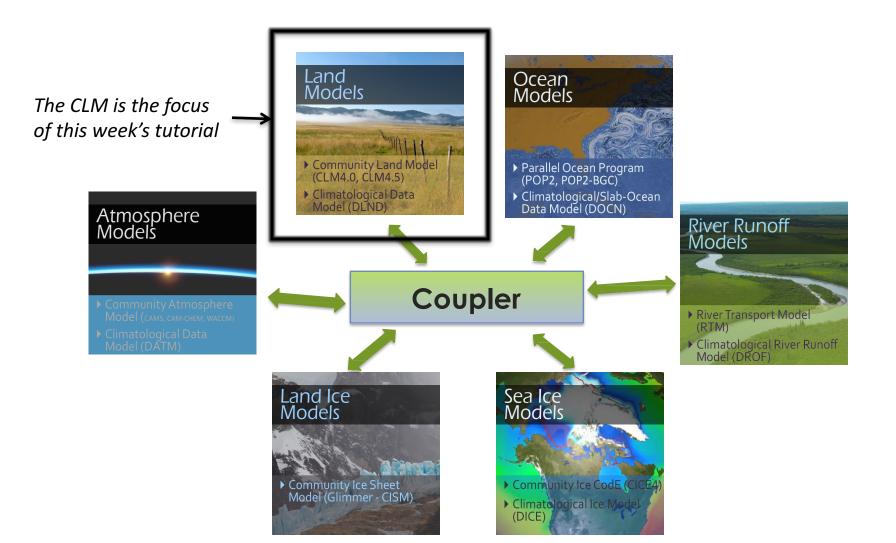
- 1) Component Sets
- 2) ENV files (env\_[command])
- 3) Namelist files (user\_nl\_[model])

### Getting help

### Appendix

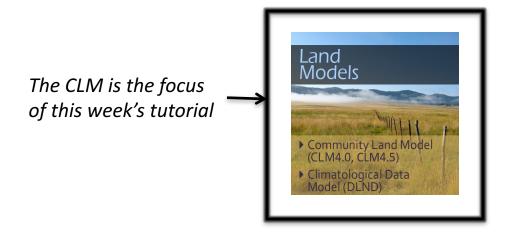
## The CESM Framework

The Community Earth System Model (CESM) is a set of models that can be run independently or together to simulate the Earth global climate.



## The CESM Framework

The Community Earth System Model (CESM) is a set of models that can be run independently or together to simulate the Earth global climate.



The CLM (and the CESM) can be run through a set of scripts provided with the model.

The first part of this practical session is a quick start to the CLM workflow (out-of-the-box)



<u>out of the box</u> = works immediately after installation without any modification

## **CESM 1.2 Web Page**

http://www.cesm.ucar.edu/models/cesm2.0/

CESM Models

Home » CESM Models » CESM2.0 Series Public Release

#### \*\* DRAFT \*\* CESM2.0 SERIES PUBLIC RELEASE

Release Notes



CESM2.0 SERIES RELEASE NOTES

#### SCIENTIFIC VALIDATION

Scientific validation

Scientific validation consists of a multi-decadal model run of the given component set at the target resolution, followed by scientific review of the model output diagnostics. All scientifically supported component sets are also accompanied by diagnostic and model output data.

What version of the model should I use?

For a scientifically supported target component set and resolution, please refer to the Scientifically Validated Configurations for that target configuration.

#### DIAGNOSTIC PACKAGES AND NAMING CONVENTIONS

- Post Processing Utilities
- Model File Naming Conventions
- Experiment Case Naming Conventions

#### MODEL DOCUMENTATION

#### CESM2.0

Model \_\_\_\_\_
Documentation

Post processing

**Tools** 

- -51-1L.0
- User's GuideMachines
- Resolutions
- Component Sets
- Model Component Namelists
- \$CASEROOT xml files

#### Component Models

- Atmosphere Models
- Land Models
- Sea Ice Models
- Ocean Models
- · Land Ice Models
- River Models
- Coupler
- ---

#### EXTERNAL LIBRARY DOCUMENTATION

- Parallel I/O Library (PIO)
- Model Coupling Toolkit (MCT)
- Earth System Modeling Framework (ESMF)

### Model Input Data



The input data necessary to run all supported component sets is made available from a public Subversion input data repository. Note that the input data repository has much more data in it than you need to run CESM2.0 — DO NOT attempt to svn checkout the whole input data repository. The CESM2.0 User's Guide explains how to obtain the subset of input data required for your needs.



#### PERFORMANCE AND LOAD BALANCING DATA

The development and testing of CESM2.0 required several iterations in the CESM1.Y series. The following timing tables for each model revision are available:

- · CESM1.3 Timing Table
- CESM1.5 Timing Table
- CESM2.0 Timing Table

#### **CESM PROJECT**

The Community Earth System Model (CESM) is a fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.

CESM is sponsored by the National Science Foundation (NSF) and the U.S. Department of Energy (DOE). Administration of the CESM is maintained by the Climate and Global Dynamics Laboratory (CGD) at the National Center for Atmospheric Research (NCAR).



### MODEL SOURCE CODE

### Copyright and Terms of Use

All CESM source code is subject to the following Copyright Notice and Disclaimer.

### Acquiring the Release Code

The source code for CESM releases is distributed through a public Subversion code repository. This code can be checked out using Subversion client software, such as the command tool svn, or simply view the latest version with a web prowser.

A short registration is required to access the repository. After registering, you will receive an email containing a user name and password that is necessary to gain access to the repository.

Acquistion of the code is more fully described in the most recent version of the CESM1.2 User's Guide.

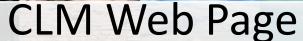
#### REPORTING A PROBLEM

If you have any problems, please first read the User's Guide including the sections on FAQs and Use Cases. Please also refer to the CESM Bulletin Board, which is in place to facilitate communication within the CESM community. Finally, please also refer to the Release Notes entries that are provided with every release and release update. If questions or problems still exist, then please send an email to cesm-help@cgd.ucar.edu. Support questions will be answered as resources are available.

CESM SUPPORT POLICY

## How to acquire the code

Reporting problems
Getting Help



http://www.cesm.ucar.edu/models/cesm1.2/clm



Home » CESM Models » CESM1.2 Public Release » CESM1.2: CLM Documentation

#### **CESM1.2: CLM DOCUMENTATION**

#### INTRODUCTION

The Community Land Model versions 4.0 and 4.5 in CESM1.2.0 are the latest in a series of land models developed through the CESM project. More information on the CLM project and access to previous CLM model versions and documentation can be found via the CLM Web Page.



### Model Documentation



#### DOCUMENTATION

- User's Guide for CLM4.5 and CLM4.0 in CESM1.2.0 [html] (Last update: Jul/20/2013)
- Technical Description for CLM4.5 (Last update: Aug/ 1/2013)
- Technical Description for CLM4.0, CLM4.0 Urban Model, CLM4.0 Crop and Irrigation Model
- Explanation of supported configurations in CLM4.5 and CLM4 in CESM1.2
- What's new in CLM in CESM1.2 (CLM4.5 release) Science, CESM1.2 (CLM4.5 release) Software, CESM1.1.1, CESM1.1.0, CESM1.0.5, CESM1.0.4, CESM1.0.3, CESM1.0.2, CESM1.0.1, CESM1.0, CCSM4.0 (CLM4.0 release).
- Known bugs in CLM in CESM1.2.0, CESM1.1.0, CESM1.0.4, CESM1.0.3, CESM1.0.2, CESM1.0.1, CESM1.0.
- Known limitations in CLM in CESM1.2.0. CESM1.1.0.

### Model Diagnostics



#### MODEL OUTPUT AND OFFLINE FORCING DATA AND DIAGNOSTIC PLOTS

- CLM4.0 and CLM4.5 offline control simulations: Diagnostic plots
- CLM4.0 and CLM4.5 offline control simulations (links need to be updated and data posted to ESG)): Model output data
- CLM4.0 and CLM4.5 offline control simulations (links need to be updated and data posted to ESG): Model forcing data
- CLM4.0 and CLM4.5 offline historical and RCP simulations: CCSM4 coupler history forcing data

### **Post-processing Tools**



#### **CLM POST-PROCESSING AND ANALYSIS UTILITIES**

- · CLM Diagnostic Package: Introduction, Code (via syn repository, registration required), and User's Guide
- Multivariate visual analytics tool: EDEN (Exploratory Data analysis Environment)
   Helps reveal associations among variables for guided analysis (beta version, comments to Chad Steed)

### MODEL DESIGN AND DEVELOPMENT

- Request Form for Developer Access (active close collaborators only)
- CLM Developers' Guidelines
- CLM4/CESM1.2.0 Code Reference Guide

#### REFERENCES

Bibliography of papers utilizing and/or developing CLM (Last update: Jan/29/2016)

#### CIM4 5

 Oleson, K.W., D.M. Lawrence, G.B. Bonan, B. Drewniak, M. Huang, C.D. Koven, S. Levis, F. Li, W.J. Riley, Z.M. Subin, S.C. Swenson, P.E. Thornton, A. Bozbiyik, R. Fisher, E. Kluzek, J.-F. Lamarque, P.J. Lawrence, L.R. Leung, W. Lipscomb, S. Muszala, D.M. Ricciuto, W. Sacks, Y. Sun, J. Tang, Z.-L. Yang, 2013: Technical Description of version 4.5 of the Community Land Model (CLM). Ncar Technical Note NCAR/TN-503+STR, National Center for Atmospheric Research, Boulder, CO, 422 pp, DOI: 10.5065/D6RR1W7M.

CRUNCEP Forcing (standard for CLM4.5, but can also be used for CLM4.0

 Piao et. al. 2012: The carbon budget of terrestrial ecosystems in East Asia over the last two decades. Biogeosciences, 9, 3571-3586, doi:10.5194/bq-9-3571-2012. (CRUNCEP forcing)

#### CLM4.0

- Lawrence, D.M., K.W. Oleson, M.G. Flanner, P.E. Thornton, S.C. Swenson, P.J. Lawrence, X. Zeng, Z.-L. Yang, S. Levis, K. Sakaguchi, G.B. Bonan, and A.G. Slater, 2011: Parameterization improvements and functional and structural advances in version 4 of the Community Land Model. J. Adv. Model. Earth Sys., 3, DOI: 10.1029/2011MS000045.
- Oleson, K.W., D.M. Lawrence, G.B. Bonan, M.G. Flanner, E. Kluzek, P.J. Lawrence, S. Levis, S.C. Swenson, P.E. Thornton, A. Dai, M. Decker, R. Dickinson, J. Feddema, C.L. Heald, F. Hoffman, J.-F. Lamarque, N. Mahowald, G.-Y. Niu, T. Qian, J. Randerson, S. Running, K. Sakaguchi, A. Slater, R. Stockli, A. Wang, Z.-L. Yang, Xi. Zeng, and Xu. Zeng, 2010: Technical Description of version 4.0 of the Community Land Model (CLM). NCAR Technical Note NCAR/TN-478+STR, National Center for Atmospheric Research, Boulder, CO, 257 pp.

## **CESM Directory Structure**

**CESM Source Code** 

**CESM** data

**\$CCSMROOT** 

**\$DIN LOC ROOT** 

### **Code & Data directories:**

### - CESM source code

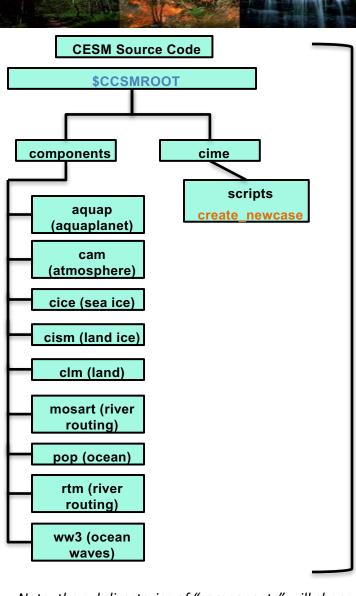
/glade/p/cesm/lmwg/CLM2016\_Tutorial\_cesm2\_0\_beta01

**Note**: Today we are using a pre-released version of CESM2.0. Released CESM code is in: /glade/p/cesm/releases

### - CESM data

/glade/p/cesm/cseg/inputdata

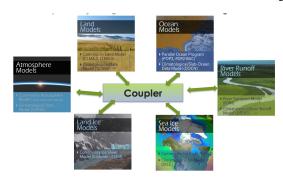
## **CESM Directory Structure**



\$DIN\_LOC\_ROOT

### Source code has 2 subdirectories:

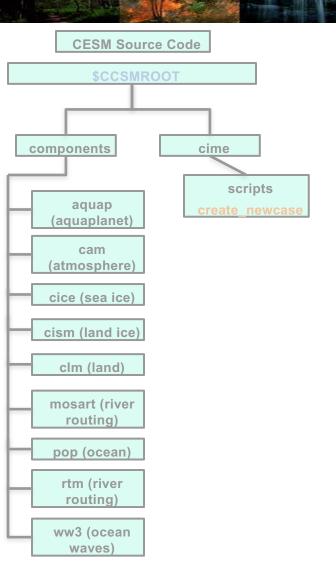
- components: contains the code for every model component

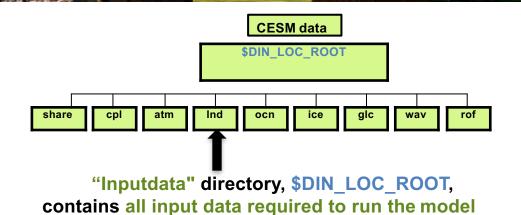


- cime: contains the scripts you need to run CESM

Note: the subdirectories of "components" will change based on whether you are using a CESM code base or a CLM code base. Shown here is the structure for the CESM model base. For CLM, you will only find "clm", "mosart" or "rtm", and "cism" subdirectories. You can use either code base to run CLM-only simulations, which are defined by the component set you choose. More information on component sets is presented later.

## **CESM Directory Structure**





# Outline

### • CESM at a glance

- 1) The CESM framework
- 2) Finding information about CLM & CESM
- 3) Overview of CLM (and CESM) directory structure



### Basic workflow

- 1) Create a new case
- 2) Invoke case\_setup
- 3) Build the executable
- 4) Run and output data

### Changing configuration options

- 1) Component Sets
- 2) ENV files (env\_[command])
- 3) Namelist files (user\_nl\_[model])

### Getting help

### Appendix



## **CESM** can be run in 4 steps:

### (1) create a new case

This step sets up a new simulation. It is the most complicated of these four steps because it involves making choices to set up the model configuration

## (2) invoke case.setup

This step configures the model so that it can compile

## (3) build the executable

This step compiles the model

## (4) submit your run to the batch queue

This step submits the model simulation to the supercomputer queue

Here, you will learn to use these four steps to set up and run a simulation. After completing this section, you will learn how to make basic modifications to the model configuration.



## **CESM** can be run in 4 steps:



### (1) create a new case

This step sets up a new simulation. It is the most complicated of these four steps because it involves making choices to set up the model configuration

## (2) invoke case.setup

This step configures the model so that it can compile

## (3) build the executable

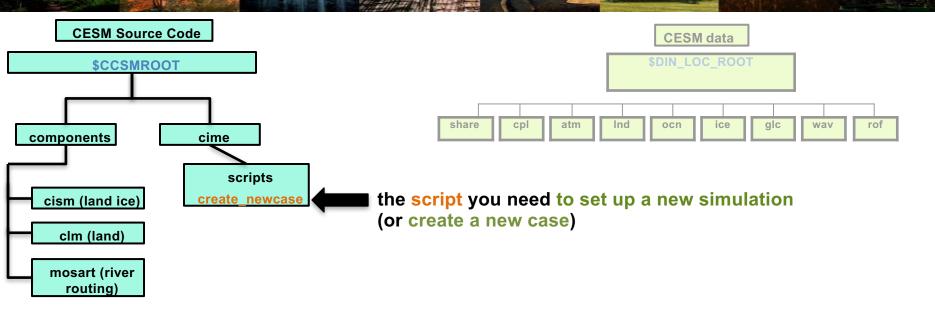
This step compiles the model

## (4) submit your run to the batch queue

This step submits the model simulation to the supercomputer queue

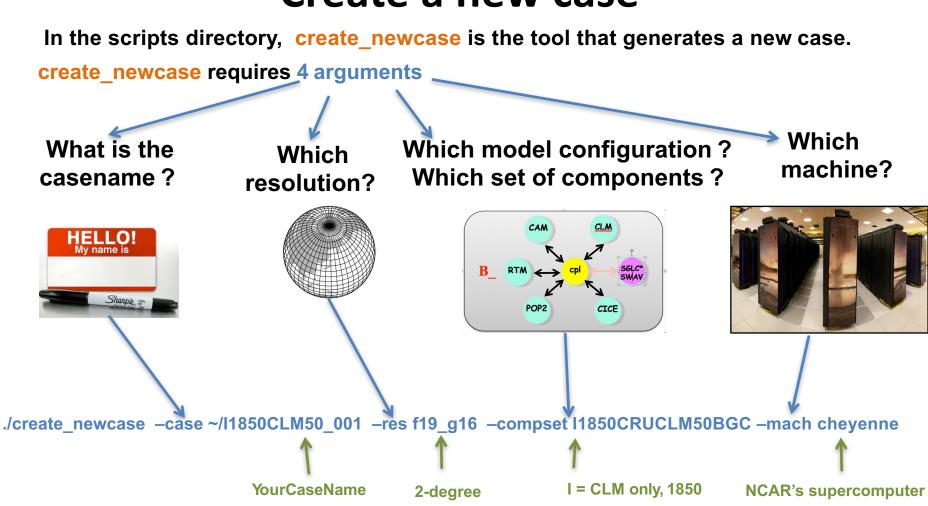
Here, you will learn to use these four steps to set up and run a simulation. After completing this section, you will learn how to make basic modifications to the model configuration.

## **CLM Directory Structure**



Note: This week, we are using a CLM code base, which has fewer subdirectories than the CESM code base.

## Create a new case





4 arguments required by create\_newcase:

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

4 arguments required by create\_newcase:

Recommendation: Use meaningful names, including model version, type of simulation, and any additional details to help you remember the configuration of this simulation

Note: Steps 2-4 take place in the case directory that you create here in step 1. More on that coming up.

4 arguments required by create\_newcase:

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

res specifies the model resolutions (or grid): f19\_g16 (atm/Ind\_ocn/ice)

**Grid naming convention** 

Each model resolution can be specified by its alias, short name and long name.

**Example of equivalent alias, short name and long name:** 

- alias: f19\_g16 (atm/Ind\_ocn/ice)
- short name: 1.9x2.5\_gx1v6

4 arguments required by create\_newcase:

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

compset specifies the "component set"

Component set specifies component models (e.g. active vs data), forcing scenarios (e.g. 1850 vs 2000) and physics options (e.g. CLM4.5 vs CLM5.0) for those models. All CLM-only compsets start with "I".

### **Compset naming convention**

Each model compset can be specified by its alias, short name and long name. Example of equivalent alias, short name and long name:

alias: I1850CRUCLM50BGC





4 arguments required by create\_newcase:

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

mach specifies the machine (or computer) name

The computer you are using will determine the syntax of the commands you use.

Note: The example syntax used throughout this presentation is written as used on NCAR's computers

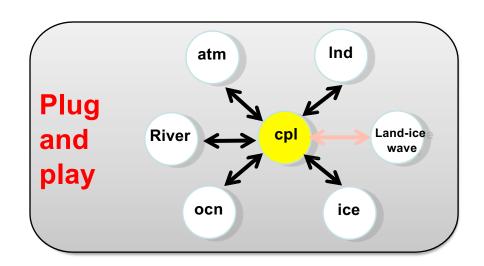
## More on CESM component sets

Plug and play of components with different component models

Color code:



stub



### **Key Definitions:**

Active: Simulation is using the code from the model during the run

Simulation is reading in data from a file for this component Data:

Stub: Component is not being used

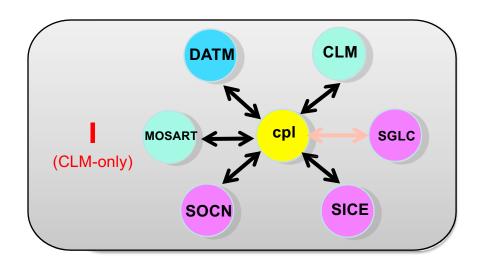
## More on CESM component sets

Plug and play of components with different component models

Color code: acti



stub



### **Key Definitions:**

**Active**: Simulation is using the code from the model during the run

**Data**: Simulation is reading in data from a file for this component

**Stub**: Component is not being used

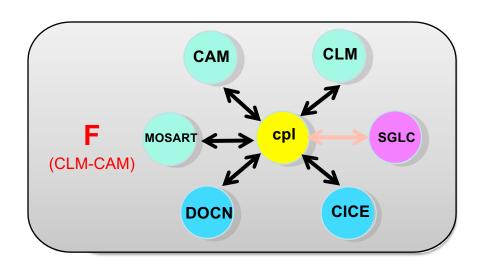
## More on CESM component sets

Plug and play of components with different component models

Color code: activ



stub



### **Key Definitions:**

**Active**: Simulation is using the code from the model during the run

**Data**: Simulation is reading in data from a file for this component

**Stub**: Component is not being used



# create\_newcase: More Information & Help

In the scripts directory (in the Source Code), where you run the command "create\_newcase", you can search for compsets, resolutions (e.g. model grid), etc.:

```
./manage_case --query <type>
```

Where "type" can be: compsets -setby <name>; component -name <name>; grids; machines And "name" is the name of a particular model component, like clm, cam, datm, etc.

### For example:

./manage\_case --query compsets --setby clm
Will list all the "I" compsets available

For additional help and options: ./manage\_case --help



## create newcase: More Information & Help

Home » CESM Models » CESM2.0 Series Public Release

### \*\* DRAFT \*\* CESM2.0 SERIES PUBLIC RELEASE

#### **ABOUT THIS RELEASE SERIES**

#### **CESM2.0 SERIES RELEASE NOTES**

#### SCIENTIFIC VALIDATION

Scientific validation consists of a multi-decadal model run of the given component set at the target resolution, followed by scientific review of the model output diagnostics. All scientifically supported component sets are also accompanied by diagnostic and model output data.

#### What version of the model should I use?

For a scientifically supported target component set and resolution, please refer to the Scientifically Validated Configurations for that target configuration.

#### DIAGNOSTIC PACKAGES AND NAMING CONVENTIONS

- Post Processing Utilities
- Model File Naming Conventions
- Experiment Case Naming Conventions

#### MODEL DOCUMENTATION

### CESM2.0

- · User's Guide
- Machines
- Resolutions
- Component Sets Model Component Namelists
- \$CASEROOT xml files

#### Component Models

- Atmosphere Models
- Land Models Sea Ice Models
- Ocean Models
- Land Ice Models
- River Models Coupler

#### CESM PROJECT

The Community Earth System Model (CESM) is a fully-coupled, global climate model that provides state-ofthe-art computer simulations of the Earth's past, present, and future climate states.

CESM is sponsored by the National Science Foundation (NSF) and the U.S. Department of Energy (DOE). Administration of the CESM is maintained by the Climate and Global Dynamics Laboratory (CGD) at the National Center for Atmospheric Research (NCAR).

### MODEL SOURCE CODE Copyright and Terms of Use

All CESM source code is subject to the following Copyright Notice and

#### Acquiring the Release Code

The source code for CESM releases is distributed through a public Subversion code repository. This code can be checked out using Subversion client software, such as the command tool syn, or simply view

A short registration is required to access the repository. After registering, you will receive an email containing a user name and password that is necessary to gain access to

### A list of valid values is also available on the CESM website:

http://www.cesm.ucar.edu/models/cesm2.0/



### Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

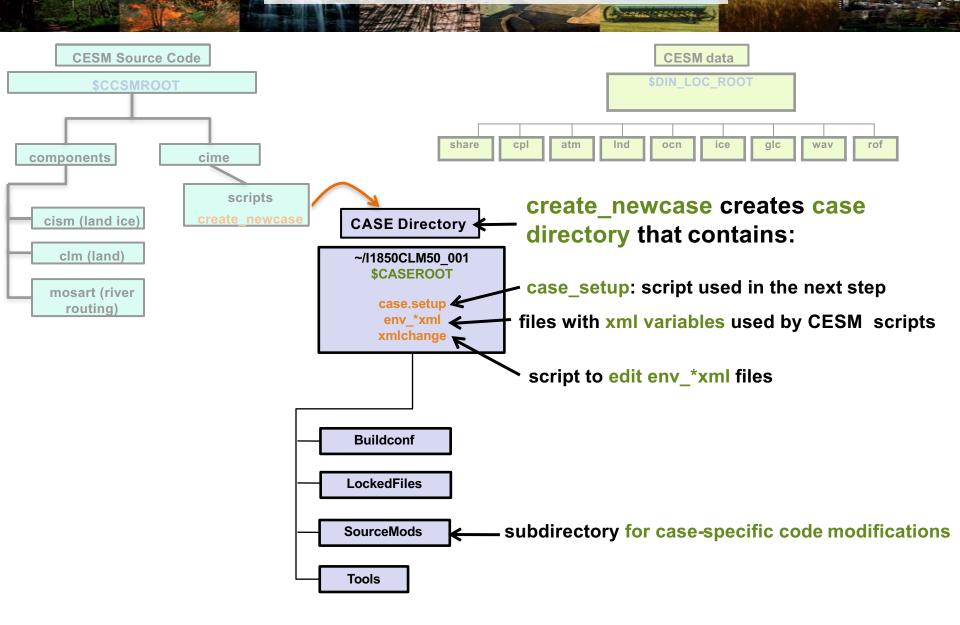
cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne
This command line creates a case directory with the case name you specified. Let's explore that directory structure.

- (2) invoke case.setup
- (3) build the executable
- (4) submit your run to the batch queue

## **CLM Directory Structure**





## Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne
Using this command line, we just set up a new simulation and created the case directory.

(2) invoke case.setup

(3) build the executable

(4) submit your run to the batch queue



## Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

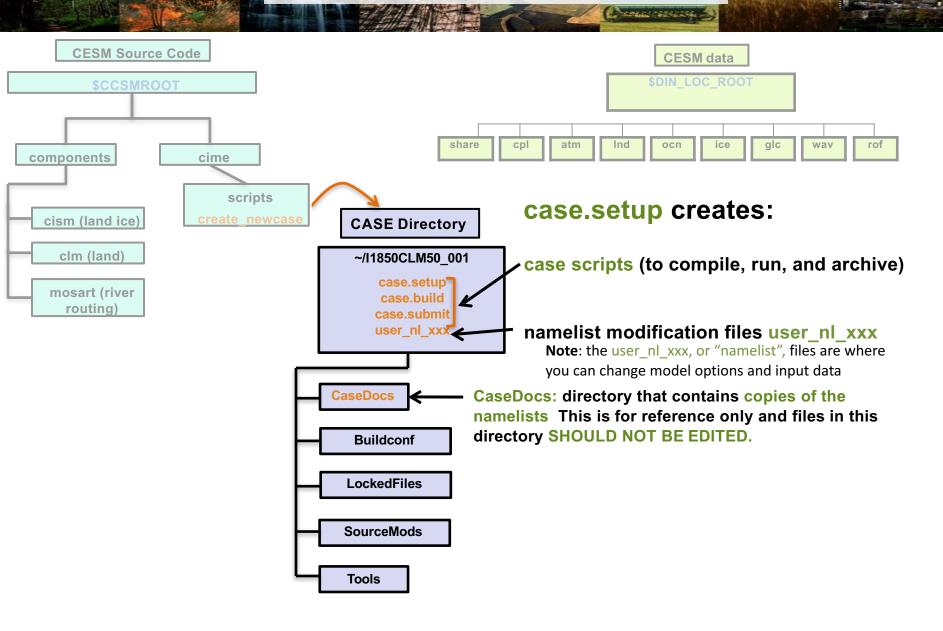


(2) invoke case.setup

Now we'll configure the case you just set up.

- (3) build the executable
- (4) submit your run to the batch queue

## **CLM Directory Structure**





## Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001

./case.setup

Using this command line, we just configured the model and created the files to modify options & input data.

(3) build the executable

(4) submit your run to the batch queue



## Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001 ./case.setup



(3) build the executable

Next, we will compile the model code

(4) submit your run to the batch queue

### **CLM Directory Structure CESM Source Code CESM** data \$DIN LOC ROOT **\$CCSMROOT** share cpl atm Ind ocn ice glc components cime The case.build script: scripts Checks input data cism (land ice) **Build/Run Directory CASE Directory** Creates a atm clm (land) ~/I1850CLM50 001 **\$EXEROOT** build/run directory with model Ind mosart (river executable and routing) ocn namelists ice CaseDocs glc bld cpl Buildconf wav LockedFiles lib SourceMods run **Tools \$RUNDIR**

Note: If an input data is missing, build aborts and provides a list of missing files.



### Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001
./case.setup

(3) build the executable

Type this command line:

./case.build

Using this command line, we just compiled the model and created a run directory with model executables.

(4) submit your run to the batch queue



## Exercise 1: Create & run an out-of-the-box simulation

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001
./case.setup

(3) build the executable

Type this command line:

./case.build

(4) submit your run to the batch queue

We're on the last step! We will submit the simulation to the supercomputer queue



## **Submit and Check Job Status**

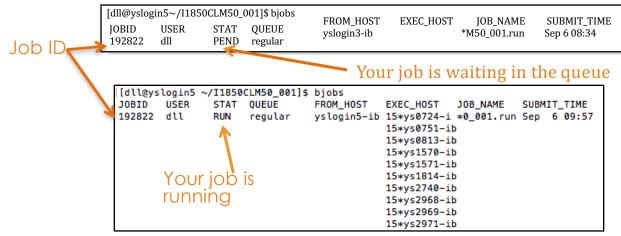
When you submit a job, you will see confirmation that it successfully submitted:

```
Check case OK
submit_jobs case.run
job is case.run
Submit job case.run
Submit job case.run
Submitting job script bsub -q regular -W 12:00 -P P93300641 < case.run
[dll@yslogin3 ~/I1850CLM50_001]$ [
```

Your job was submitted to the regular queue ("q"). Wallclock time ("W") and project number ("P") are also specified

### **Checking jobs:**

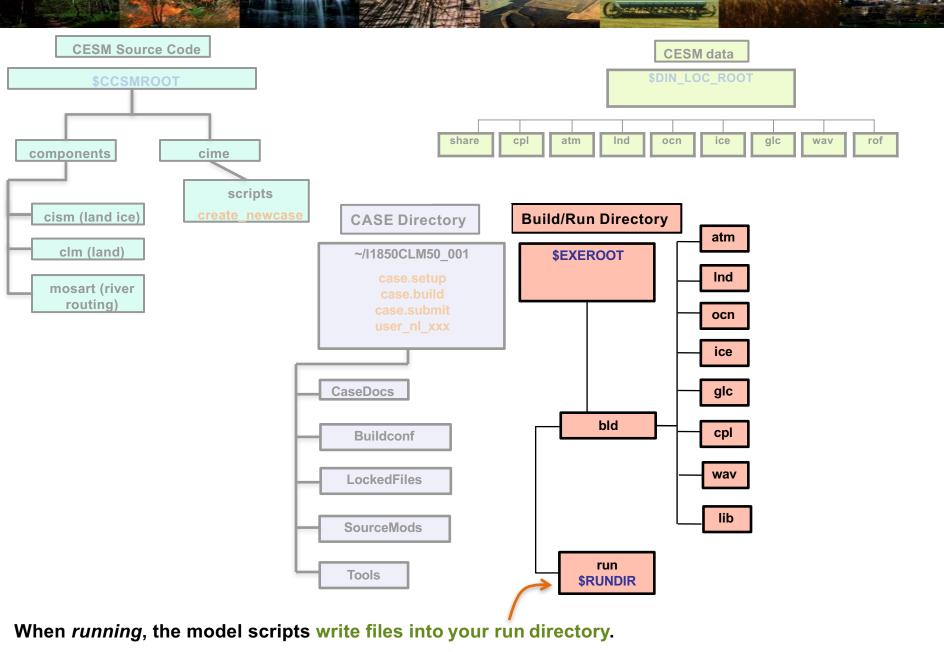
- a. Type *qstat –u {user-id}* to see your jobs or
- b. Type *qstat* to see everyone's jobs



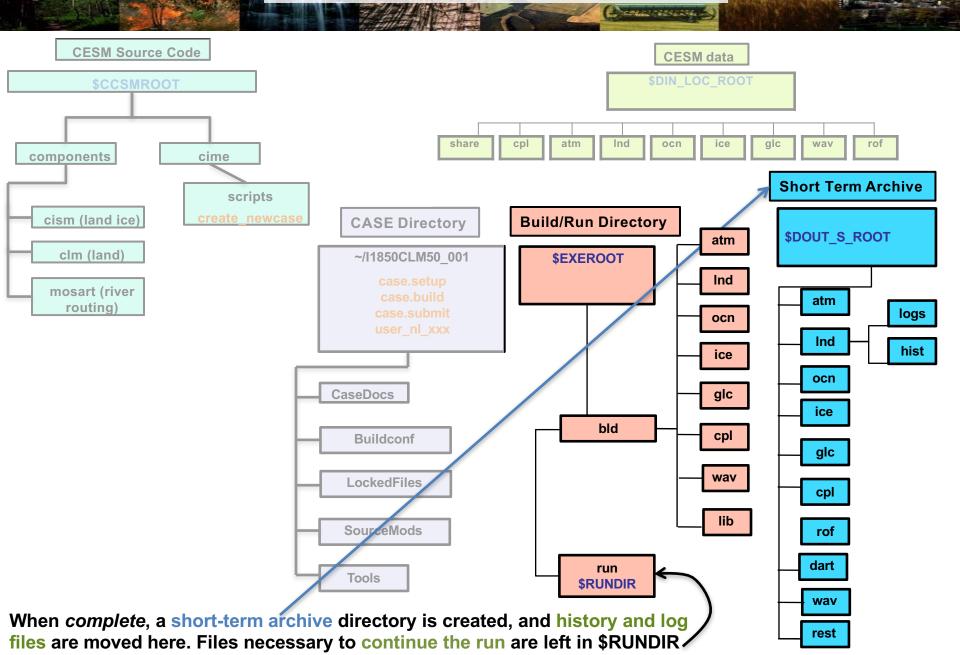
### Killing jobs:

- a. Find your Job ID after typing *qstat –u {user-id}*
- b. Type *qdel <Job ID>*

## **CLM Directory Structure**



## **CLM Directory Structure**





# Up Next: Making changes to CLM configuration options

# Outline

#### • CESM at a glance

- 1) The CESM framework
- 2) Finding information about CLM & CESM
- 3) Overview of CLM (and CESM) directory structure

#### Basic workflow

- 1) Create a new case
- 2) Invoke case\_setup
- 3) Build the executable
- 4) Run and output data

#### Changing configuration options

- 1) Component Sets
- 2) ENV files (env\_[command])
- 3) Namelist files (user\_nl\_[model])

#### Getting help

#### Appendix



#### Review: The 4 commands to run CLM

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001
./case.setup

(3) build the executable

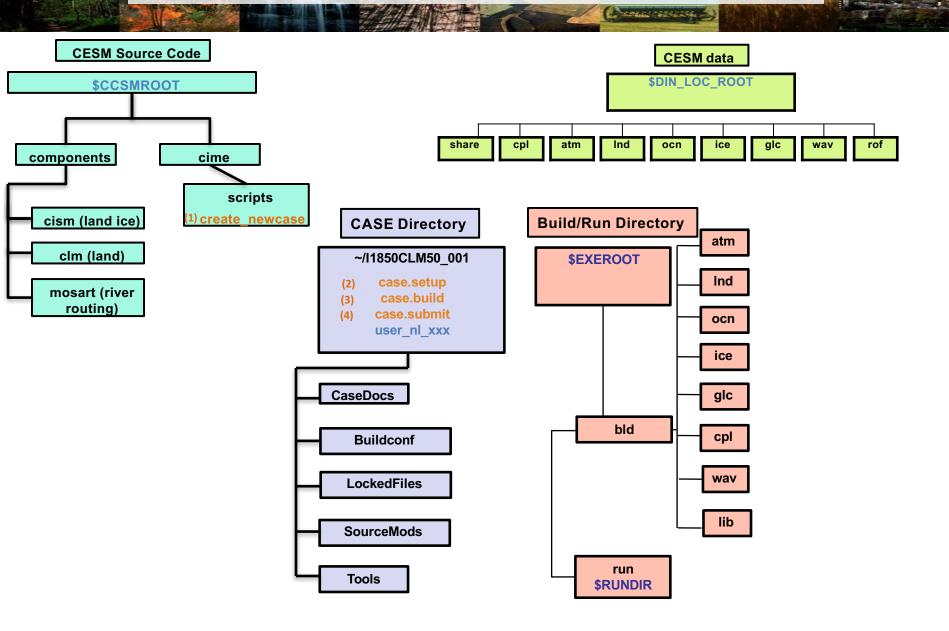
Type this command line:

./case.build

(4) submit your run to the batch queue

Type this command line:

./case.submit





#### **Review: Queues and Jobs**

#### On Yellowstone

#### **Checking jobs:**

- a. Type *qstat –u dll* to see your jobs or
- b. Type *qstat* to see all jobs

#### Killing jobs:

- a. Find your JOBID after typing *qstat*
- b. Type *qdel <JOBID>*



# Finding model output

Directory:

Change this to your user name

/pathtorundir/{userXX}/archive/I850CLM50\_001/Ind/hist



# Finding model output

Directory: Change this to your user name

/ pathtorundir/{userXX}/archive/I850CLM50\_001/Ind/hist

Files (use "Is" to list them):

I1850CLM50\_001.h0.0001-12.nc

Case Name

Time

Output Type (history)

File Type (netCDF)

# 3 Types of Basic Modifications

1. Component Sets

ENV files (env\_[command])

3. Namelist files (user\_nl\_[model])



# 3 Types of Basic Modifications

1. Component Sets

Set up a simulation for 2000

2. ENV files (env\_[command])

Namelist files (user\_nl\_[model])



#### Create a new case

In the scripts directory, **create\_newcase** is the tool that generates a new case.

create\_newcase requires\_4 arguments

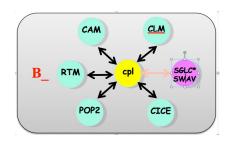
What is the casename?

Which resolution?

Which model configuration? Which set of components?

Which machine?







Example used:

11850CLM50\_001

f19\_g16 (2-degree)

I1850CRUCLM50BGC (I = CLM only, 1850) Cheyenne (NCAR supercomputer)

The command line:

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne



#### Create a new case

In the scripts directory, **create\_newcase** is the tool that generates a new case.

create newcase requires 4 arguments

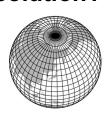
What is the casename?

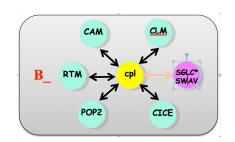
Which resolution?

Which model configuration? Which set of components?

Which machine?









I1850CLM50\_001

f19\_g16 (2-degree)

I1850CRUCLM50BGC (I = CLM only, 1850) Cheyenne (NCAR supercomputer)

Now we'll change the compset

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne



# **Changing Simulation Components**

Compset, or component set: predefined options for running the model

Use compset to change the type of simulation



#### Changing compsets lets you run different experiments

#### Some component options:

- Year (1850, 2000, transient, etc.)
- Data atmosphere (GSWP3, CRUNCEP, CPLHIST3HrWx)
- Model options (SP [satellite phenology], BGC [biogeochemistry])
- RCP scenarios



#### Changing compsets lets you run different experiments

#### Some component options:

- Year (1850, 2000, transient, etc.)
- Data atmosphere (GSWP3, CRUNCEP, CPLHIST3HrWx)
- Model options (SP [satellite phenology], BGC [biogeochemistry])
- RCP scenarios

#### **Examples of simulations using different compsets:**

- Stabilize ("spin up") a biogeochemsitry (includes N & C cycles) simulation for 1850
- Run a transient historical simulation from 1850-2000 based on the 1850 spin up
- Run a transient future simulation from 2000 through 2100 using RCP8.5
- Run a "time slice" simulation for 2000



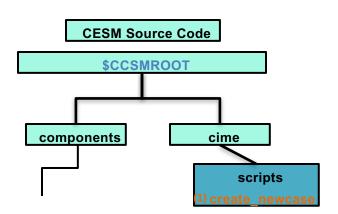
## Where to find a list of compsets:

http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/modelnl/compsets.html

Website lists ALL compsets for released CESM. CLM only = "I" compsets

#### In CESM scripts directory, can run:

./manage\_case --query compsets --setby clm



Tip: Add " | more" at the end of the command line, then use the spacebar to scroll through the options



#### Create a new case

In the scripts directory, **create\_newcase** is the tool that generates a new case.

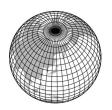
create newcase requires 4 arguments

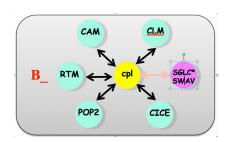
What is the casename?

Which resolution? Which model configuration? Which set of components?

Which machine?









12000CLM50\_001

f19 g16 (2-degree) ICRUCLM50BGC

Cheyenne (I = CLM only, 2000) (NCAR supercomputer)

./create newcase -case ~/I2000CLM50 001 -res f19 g16 -compset ICRUCLM50BGC -mach cheyenne



#### **Exercise 2: Create & build simulation for 2000**

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I2000CLM50\_001 -res f19\_g16 -compset ICRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I2000CLM50\_001 ./case.setup

(3) build the executable

Type this command line:

./case.build

(4) submit your run to the batch queue

Type this command line:

./case.submit



# 3 Types of Basic Modifications

1. Component Sets

2. ENV files (env\_[command].xml)

Changing the length of the run

3. Namelist files (user\_nl\_[model])



## **Exercise 3: Change the length of simulated time**

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I2000CLM50\_001 -res f19\_g16 -compset ICRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I2000CLM50\_001 ./case.setup

(3) build the executable

Type this command line:

./case.build

Change the run length BEFORE submitting

(4) submit your run to the batch queue

Type this command line:

./case.submit



## How To: Change the length of simulated time

Two methods of changing the run length. Method 1 uses the "xmlchange" script and is the preferred method

#### Use when modifying "xml" files (e.g. env\_run.xml)

- Benefit: Won't let you mess up the syntax!
- 2. For help, type ./xmlchange -help
- 3. Use "./xmlquery list" to list all variables and their values in all the .xml files

Example: editing env\_run.xml via the xmlchange tool

./xmlchange {variable to be changed}={value to change to}

<sup>\*</sup> We won't use xml commands right now, but you will during the next section.



## How To: Change the length of simulated time

Method 2 edits the scripts directly and involves two steps:

- 1) Modify env\_run.xml to set desired simulated length (STOP\_N and STOP\_OPTION)
- 2) Modify env\_batch.xml to tell computer how much computer time is needed to complete the simulation (JOB\_WALLCLOCK\_TIME)

When modifying files, use an editor of your choice

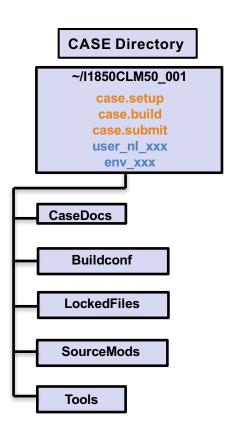
Examples:

Emacs nedit



## **Exercise 3: Change the length of simulated time**

1)Modify env\_run.xml to set desired simulated length



In a text editor\*, open env\_run.xml

\* If you don't have a preferred editor, emacs is user friendly. Type "emacs env run.xml" (or "emacs anyfilename")

## env\_run.xml script

#### This is the beginning of the env\_run.xml script:

```
<config definition>
<header>
   These variables MAY BE CHANGED ANYTIME during a run.
   Additional machine speific variables that can be changed
   during a run are contained in the env mach specific file
   Note1: users SHOULD NOT modify BUILD_COMPETE in env_build.xml
   this is done automatically by the scripts.
   </header>
<groups>
  <group>external_tools
  <group>run_begin_stop_restart
  <group>run_budgets</group>
  <qroup>run_cesm</group>
  <group>run_co2</group>
  <group>run_component_cism</group>
  <group>run_component_clm
  <group>run_component_cpl</group>
  <group>run_component_datm</group>
  <group>run_component_rtm</group>
  <group>run_coupling</group>
  <group>run_data_archive</group>
  <qroup>run_desc</qroup>
  <group>run_din</group>
  <group>run_domain</group>
  <group>run_dout</group>
  <group>run_drv_history</group>
  <group>run_flags</group>
  <group>run_glc</group>
  <group>run_machine</group>
  <group>run_mpi</group>
  <group>run_pio</group>
</groups>
 <entry id="DATA_ASSIMILATION" value="FALSE">
   <type>logical</type>
   <valid values>TRUE,FALSE</valid values>
   <group>external_tools
   <desc>Run the external tool pointed to by DATA_ASSIMILATION_SCRIPT after the model run completes</desc>
 </entry>
 <entry id="DATA_ASSIMILATION_CYCLES" value="1">
   <type>integer</type>
   <group>external_tools</group>
   <desc>Number of model run - data assimilation steps to complete</desc>
 </entrv>
                                                                                                                         Top
                                                                                                           1,1
```

# env\_run.xml script

Runtime variables can be changed in env\_run.xml at any point and control the mechanics of the run (length, resubmits, and archiving).

Common variables to change include:

- **1. STOP\_OPTION** → sets the run time interval type, i.e. nmonths, ndays, nyears
- 2. STOP\_N→ sets the number of intervals to run the model during the specified wallclock\* time.
  - \* Wallclock time is set in the env\_batch.xml file and is a measure of the actual time.
- 3. **RESUBMIT** > sets the number of times to resubmit the run



#### Exercise 3 (Part 1): Run simulation for 5 years

#### TO DO:

In the env run.xml script in your I2000CLM50 001 case, change:

- **1. STOP\_OPTION** → change to "nyears"
- 2. STOP\_N $\rightarrow$  change to "5"
- **3. RESUBMIT** → sets the number of times to resubmit the run

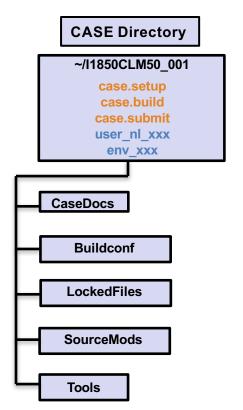
We won't use "resubmit" right now. Here is an example for how to run 5 years using the resubmit option:

$$STOP_N = 1$$
  
 $RESUBMIT = 4$ 

<sup>\*\*</sup> This will run 5 different simulations for 1 year each \*\*

## env\_batch.xml script

**Exercise 3 (Part 2)**: Modify env\_batch.xml to tell computer how much computer time is needed to complete the simulation (Wall Clock Time)



Using a text editor, open env batch.xml

# env\_batch.xml script

```
Rowsion="1.0"?>
<config_definition>
<header>
                                                                           Common Variables to change:
     These variables may be changed anytime during a run, they
     control arguments to the batch submit command.
                                   Make sure that changes are in the job "run", at the top of the script. You can also
<groups>
  <group>job_submission
                                   make changes to the short-term and long-term archive submission scripts here.
</groups>
 <iob name="run":
                                                                            .) queue type (also: economy, premium, etc.)
     <entry id="JOB_QUEUE" value="regular">
       <type>character</type>
       <group>job_submission
       <desc>The machine queue in which to submit the job. Default determined in config_machines.xml can be overwritten by testing</desc>
     </entry>
                                                                           2) wallclock time requested*
     <entry id="JOB_WALLCLOCK_TIME" value="4:00">
       <type>character</type>
       <group>job_submission</group>
       <desc>The machine wallclock setting. Default determined in config_machines.xml can be overwritten by testing</desc>
     </entry>
                                                                            3) project number
     <entry id="PROJECT" value="P93300641">
       <tvpe>char</tvpe>
       <group>job_submission</group>
       <desc>project for project-sensitive build and run paths, and job scripts</desc>
     </entry>
     <entry id="PROJECT_REQUIRED" value="TRUE">
       <type>logical</type>
       <valid_values>TRUE,FALSE</valid_values>
       <group>job_submission</group>
       <desc>whether the PROJECT value is required on this machine</desc>
     </entry>
  </iob>
 <job name="test">
     <entry id="JOB QUEUE" value="">
       <type>character</type>
       <group>job_submission</group>
       <desc>The machine queue in which to submit the job. Default determined in config_machines.xml can be overwritten by testing</desc>
     </entry>
```

\*Note: Maximum allowable wall clock time on Cheyenne is 12 hours.

Submissions requesting less time typically have shorter wait times in the queue.



#### Exercise 3 (Part ): Change wall clock time

#### TO DO:

In the env\_batch.xml script in your I2000CLM50\_001 case, change:

change JOB\_WALLCLOCK\_TIME to "2:00"



## **Exercise 3: Change the length of simulated time**

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I2000CLM50\_001 -res f19\_g16 -compset ICRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I2000CLM50\_001

./case.setup

(3) build the executable

Type this command line: ./case.build

After modifying env\_run.xml and env\_batch.xml,
Start Here

(4) submit your run to the batch queue

Type this command line:

./case.submit



# 3 Types of Basic Modifications

1. Component Sets

2. ENV files (env\_[command])

3. Namelist files (user\_nl\_[model])

\* Going back to I1850CLM50\_001 case, changing data record frequency



## **Exercise 4: Changing data record frequency**

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/I1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001
./case.setup

(3) build the executable

Type this command line:

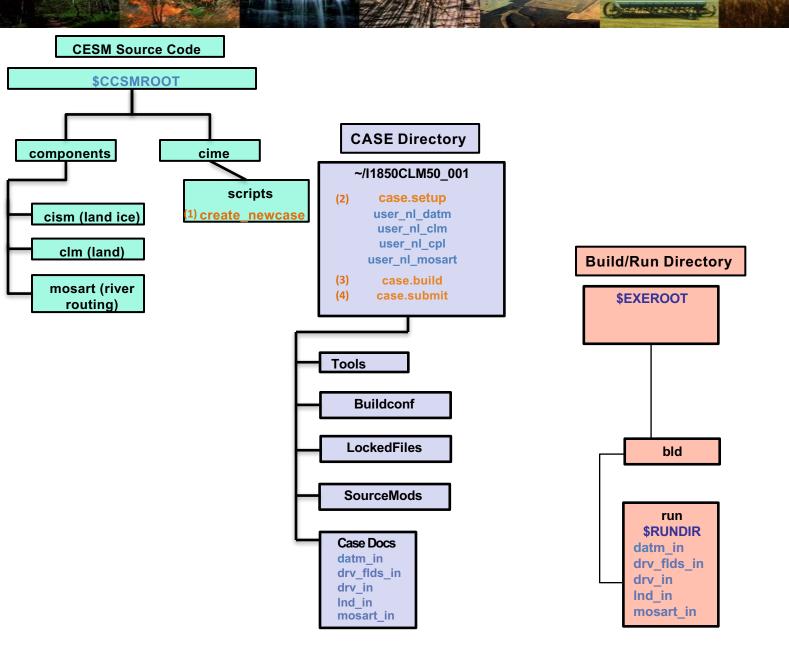
./case.build

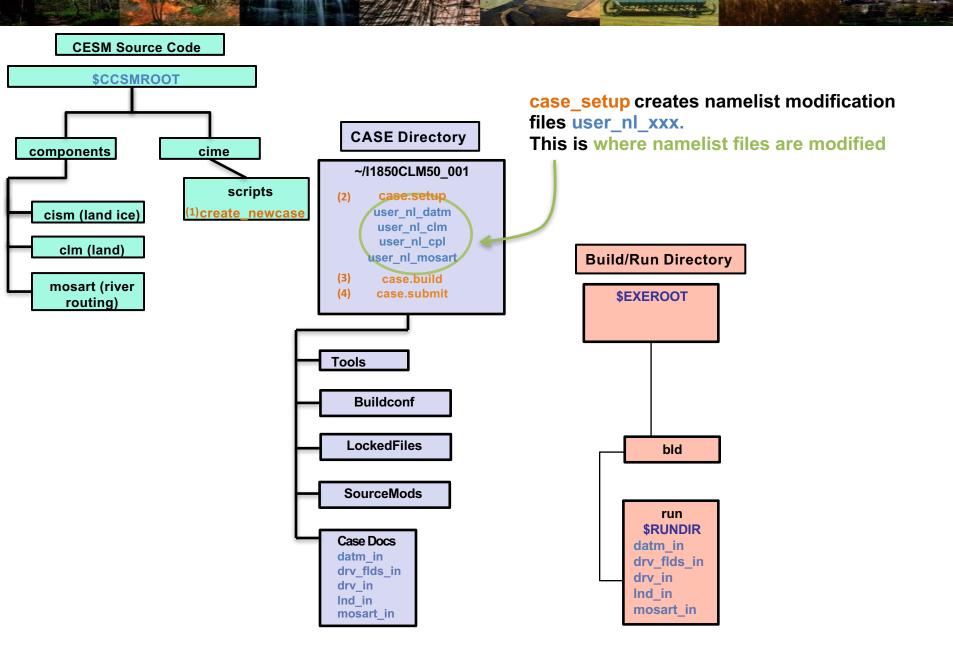
This is when you modify the namelists.

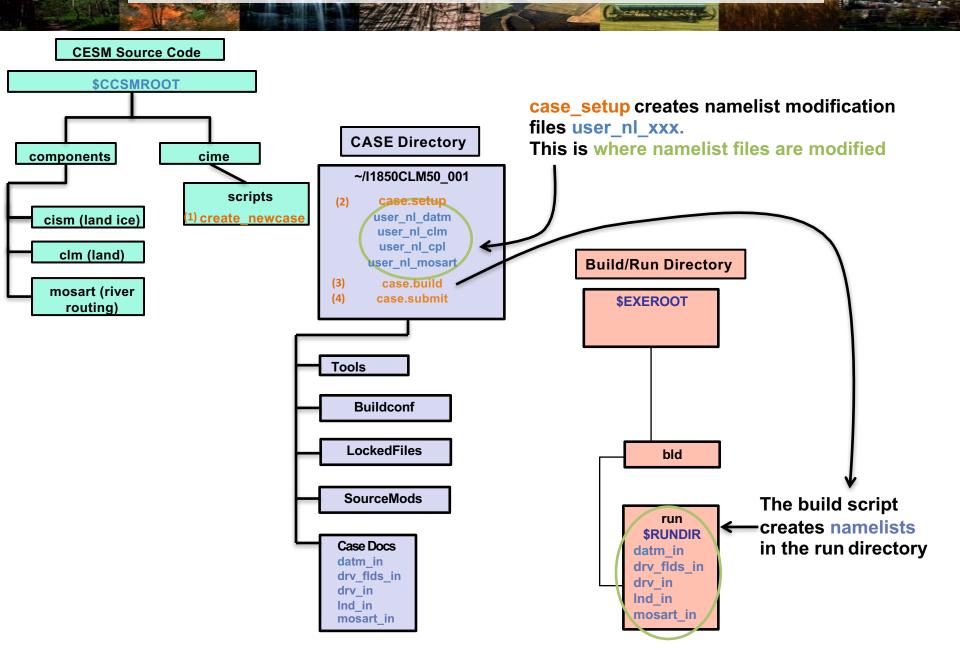
(4) submit your run to the batch queue

Type this command line:

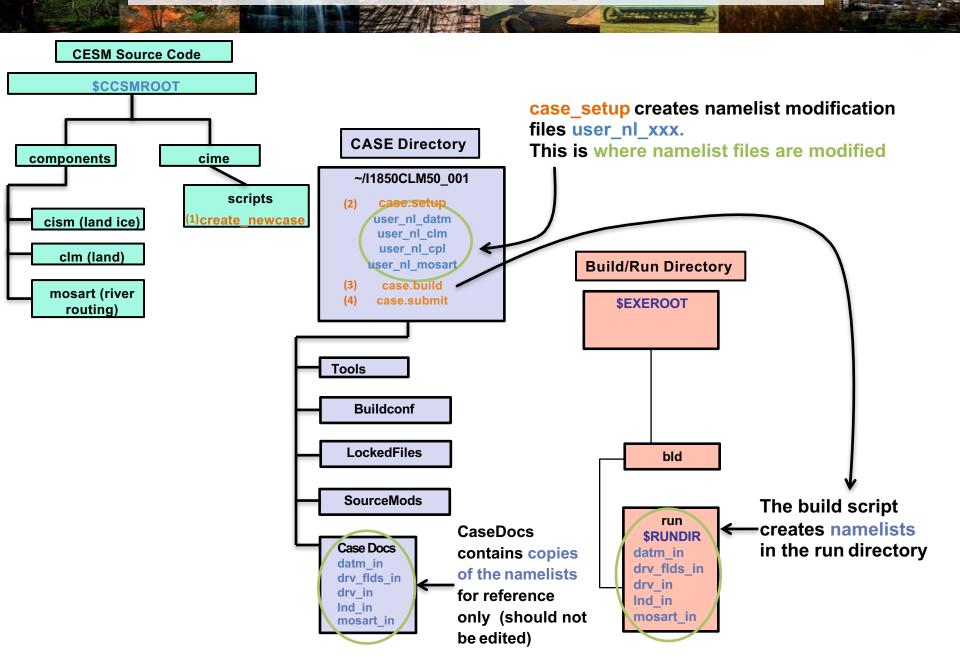
./case.submit







### Review: CLM Directories & Commands



### **Exercise 4: Changing data record frequency**

- Not all changes can be made in env run.xml.
- user\_nl\_<model> files appear in the case directory after ./case.setup is invoked:



### **Modifying Name lists**

- Compsets set up namelists
- user\_nl\_clm modifies Ind\_in name list file

Important: Don't modify the namelist file directly. Make changes in user\_nl\_clm.



### **Modifying Name lists**

- Compsets set up namelists
- user\_nl\_clm modifies Ind\_in name list file

Important: Don't modify the namelist file directly. Make changes in user nl clm.

Website for CLM namelist variables:

http://www.cesm.ucar.edu/models/cesm1.1/cesm/doc/modelnl/nl clm.html

\*\* Some namelist variables can also be changed in env\_run.xml file



# Looking at Namelist Files

Note: These files tell the input datasets and model options that your simulation is set up to use. Do not change these files directly. If changes are necessary, modify the user\_nl\_xxx files.

### Option 1

- cd into your case directory, then CaseDocs
  - (~/I1850CLM50\_001/CaseDocs)
- Open Ind\_in with text editor

### Option 2

- cd into your run directory
- Open Ind\_in with text editor

### Beginning of the Ind\_in file

```
. . .
                                         🏠 dll — dll@yslogin5:~/l1850CLM50_001 — ssh -Y dll@yellowstone.ucar.edu — 155×52
&clm_inparm
 albice = 0.60, 0.40
 co2_ppmv = 284.7
 co2_type = 'constant'
 create_crop_landunit = .false.
 dtime = 1800
 fatmlndfrc = '/glade/p/cesmdata/cseg/inputdata/share/domains/domain.lnd.fv1.9x2.5_gx1v6.090206.nc'
 finidat = ' '
 fsnowaging = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/snicardata/snicar_drdt_bst_fit_60_c070416.nc'
 fsnowoptics = '/qlade/p/cesmdata/cseq/inputdata/lnd/clm2/snicardata/snicar_optics_5bnd_c090915.nc'
 fsurdat = '/qlade/p/cesmdata/cseq/inputdata/lnd/clm2/surfdata_map/surfdata_1.9x2.5_16pfts_simyr1850_c160127.nc'
 qlc_do_dynglacier = .false.
 h2osno_max = 10000.0
 hist_mfilt = 1
 hist_nhtfrq = 0
 limit_irrigation = .true.
 maxpatch_glcmec = 0
 maxpatch_pft = 17
 nlevsno = 12
 nseqspc = 35
 paramfile = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/paramdata/clm5_params.c160713.nc'
 repartition_rain_snow = .true.
 soil_layerstruct = '20SL_8.5m'
 spinup_state = 0
 use_bedrock = .true.
 use_century_decomp = .true.
 use_cn = .true.
 use_crop = .false.
 use_dynroot = .false.
 use_ed = .false.
 use fertilizer = .false.
 use flexiblecn = .true.
 use fun = .true.
 use_grainproduct = .false.
 use_hydrstress = .true.
 use_lch4 = .true.
 use_luna = .true.
 use_nguardrail = .true.
 use_nitrif_denitrif = .true.
use_vertsoilc = .true.
&ndepdyn_nml
 ndepmapalgo = 'bilinear'
 stream_fldfilename_ndep = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/ndepdata/fndep_clm_hist_simyr1849-2006_1.9x2.5_c100428.nc'
 stream_year_first_ndep = 1850
 stream_year_last_ndep = 1850
&popd_streams
popdensmapalgo = 'bilinear'
 stream_fldfilename_popdens = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/firedata/clmforc.Li_2012_hdm_0.5x0.5_AVHRR_simyr1850-2010_c130401.nc'
 stream_year_first_popdens = 1850
```

1,1

### Beginning of the <a href="mailto:lnd">lnd</a>\_in file

```
. . .
                                         🏠 dll — dll@yslogin5:~/l1850CLM50_001 — ssh -Y dll@yellowstone.ucar.edu — 155×52
&clm_inparm
 albice = 0.60, 0.40
                                                                                                                                   [CO_2]
 co2 ppmv = 284.7
 co2_type = 'constant'
 create_crop_landunit = .false.
 dtime = 1800
 fatmlndfrc = '/glade/p/cesmdata/cseg/inputdata/share/domains/domain.lnd.fv1.9x2.5 gx1v6.090206.nc'
                                                                                                                                 File with initial conditions
 finidat = ' '
 fsnowaging = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/snicardata/snicar_drdt_bst_fit_60_c070416.nc'
 fsnowoptics = '/qlade/p/cesmdata/cseq/inputdata/lnd/clm2/snicardata/snicar_optics_5bnd_c090915.nc'
                                                                                                                                 File with surface dataset
fsurdat = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/surfdata_map/surfdata_1.9x2.5_16pfts_simyr1850_c160127.nc'
 qlc_do_dynglacier = .false.
 h2osno_max = 10000.0
 hist_mfilt = 1
 hist_nhtfrq = 0
 limit_irrigation = .true.
 maxpatch_glcmec = 0
 maxpatch_pft = 17
 nlevsno = 12
 nseqspc = 35
 paramfile = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/paramdata/clm5_params.c160713.nc'
 repartition_rain_snow = .true.
 soil layerstruct = '20SL 8.5m'
 spinup_state = 0
 use_bedrock = .true.
 use_century_decomp = .true.
 use_cn = .true.
 use_crop = .false.
 use_dynroot = .false.
 use_ed = .false.
                                        Different compsets will change the status of some of these things.
 use fertilizer = .false.
 use flexiblecn = .true.
 use fun = .true.
 use_grainproduct = .false.
 use_hydrstress = .true.
 use_lch4 = .true.
 use_luna = .true.
 use_nquardrail = .true.
 use_nitrif_denitrif = .true.
 use_vertsoilc = .true.
&ndepdyn_nml
 ndepmapalgo = 'bilinear'
 stream_fldfilename_ndep = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/ndepdata/fndep_clm_hist_simyr1849-2006_1.9x2.5_c100428.nc'
 stream_year_first_ndep = 1850
 stream_year_last_ndep = 1850
&popd_streams
popdensmapalgo = 'bilinear'
 stream_fldfilename_popdens = '/glade/p/cesmdata/cseg/inputdata/lnd/clm2/firedata/clmforc.Li_2012_hdm_0.5x0.5_AVHRR_simyr1850-2010_c130401.nc'
 stream_year_first_popdens = 1850
```

1,1

Top



## Example Modification: user\_nl\_clm

Changing the frequency of model output

hist\_mfilt: Number of samples within a file

Default is 1

Setting value to 12 would put 12 records into a single file



# Example Modification: user\_nl\_clm

Changing the frequency of model output

hist\_mfilt: Number of samples within a file

Default is 1

Setting value to 12 would put 12 records into a single file

hist\_nhtfrq: Frequency that data are recorded and written to a file

**Default**: 0 means that output is recorded every month (monthly averages)

Positive Values: Number of model timesteps (half--hourly) for output record

ex: 48 means output is recorded every day (daily averages)

Negative Values: Absolute value in hours for output record

ex: -1 means output is recorded hourly; -24 means output is recorded daily

\* Both hist\_mfilt & hist\_nhtfrq must be integers

# Example Modification: user\_nl\_clm

Changing the frequency of model output

Daily output with a years worth of daily records in a file:

$$hist_mfilt = 365$$

Monthly output with each month written to a separate file (default, as in I2000CLM50\_001 case):

For this tutorial, we changed the default data record setting to daily in the I1850CLM50 compset.

Example 4: Modify user\_nl\_clm to get monthly output, 1 file per month in I1850CLM50\_001

For this tutorial, we changed the default data record setting to daily in the I1850CLM50 compset.

Example 4: Modify user\_nl\_clm to get monthly output, 1 file per month in I1850CLM50\_001

In user\_nl\_clm, add:

hist\_mfilt = 1

hist\_nhtfrq = 0



# Then, run I1850CLM50\_001 for 5 years. To do this:

- Change user\_nl\_clm to record monthly output
- 2. Change STOP\_OPTION and STOP\_N variables in env\_run.xml
- 3. Change wall clock time in env batch.xml
- 4. Rerun the simulation: case.submit



### **Exercise 4: Changing data record frequency**

cd into scripts directory from the source code directory:

cd /{model}/{source}/{code}/cime/scripts

(1) create a new case

./create\_newcase -case ~/1850CLM50\_001 -res f19\_g16 -compset I1850CRUCLM50BGC -mach cheyenne

(2) invoke case.setup

cd into case directory:

cd ~/I1850CLM50\_001

./case.setup

(3) build the executable

Type this command line:

./case.build

After modifying the namelists, Start Here

### (4) submit your run to the batch queue

Type this command line:

./case.submit

**Note**: The case.submit script will automatically update and check the namelists. If you want to update and check your namelists before submitting, you can also run the "preview namelists" script.

### Now YOU know how to run the CLM!

Use these 3 basic modifications to run a variety of simulations.

- 1. Component Sets
- ENV files (env\_[command])
- 3. Namelist files (user\_nl\_[model])



### **Documenting Your Changes: README files**

In your case directory, you will find automatically generated documentation files.

**README.case file**: detailed information on your compset and resolution, including whether your configuration has science support.

**Best Practice:** In the *README.case file*, we highly recommend YOU document any changes you make to the default scripts. It is YOUR paper trail and opportunity to list modifications.



### **Exercises: Test Your Knowledge**

- 1) Set up a 2-degree CLM5.0-BGC simulation for 2000 and run for 1 month with daily history files.
- Set up a 2-degree CLM5.0-BGC historical simulation and run for 5 years with monthly history files
- 3) Set up a 1-degree CLM5.0-BGC 1850 simulation and run for 1 year with monthly history files



# For additional information on running & configuring CLM, see CLM User's Guide

### **CLM User's Guide:**

http://www.cesm.ucar.edu/models/cesm1.2/clm/models/lnd/clm/doc/UsersGuide/book1.html

Note: The CLM5.0 User's Guide is currently a work in progress. Look for the new version with the CESM2.0 release

### For help with other parts of the CESM:

http://www.cesm.ucar.edu/models/cesm2.0/

#### MODEL DOCUMENTATION

#### CESM2.0

- User's Guide
- Machines
- ResolutionsComponent Sets
- Model Component Namelists
- \$CASEROOT xml files

### Component Models

- Atmosphere Models
- Land Models
- Sea Ice Models
- Ocean Models
- Land Ice Models
- River ModelsCoupler
- coupici

#### EXTERNAL LIBRARY DOCUMENTATION

- Parallel I/O Library (PIO)
- Model Coupling Toolkit (MCT)
- · Earth System Modeling Framework (ESMF)

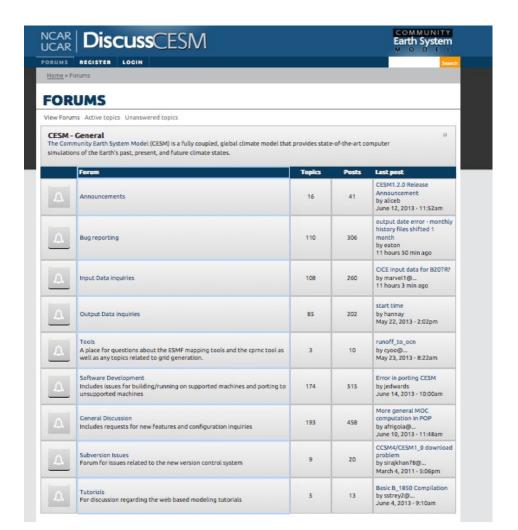
#### MODEL INPUT DATA

The input data necessary to run all supported component sets is made available from a public Subversion input data repository. Note that the input data repository has much more data in it than you need to run CESM2.0 --- DO NOT attempt to svn checkout the whole input data repository. The CESM2.0 User's Guide explains how to obtain the subset of input data required for your needs.



# Getting Help

CESM Bulletin Board: http://bb.cgd.ucar.edu/





### **Appendix**

- Registration
- Download the source code
- Hardware/software requirements

# (A) Registration

Go to CESM2.0 home page: <a href="http://www.cesm.ucar.edu/models/cesm2.0/">http://www.cesm.ucar.edu/models/cesm2.0/</a>

CESM Models

Home » CESM Models » CESM2.0 Series Public Release

#### \*\* DRAFT \*\* CESM2.0 SERIES PUBLIC RELEASE

#### **ABOUT THIS RELEASE SERIES**

#### CESM2.0 SERIES RELEASE NOTES

#### SCIENTIFIC VALIDATION

Scientific validation consists of a multi-decadal model run of the given component set at the target resolution, followed by scientific review of the model output diagnostics. All scientifically supported component sets are also accompanied by diagnostic and model output data.

#### What version of the model should I use?

For a scientifically supported target component set and resolution, please refer to the Scientifically Validated Configurations for that target configuration.

#### DIAGNOSTIC PACKAGES AND NAMING CONVENTIONS

- Post Processing Utilities
- Model File Naming Conventions
- Experiment Case Naming Conventions

### MODEL DOCUMENTATION

### CESM2.0

- User's Guide
- Machines
- Resolutions
   Component Sets
- Model Component Namelists
- SCASEROOT xml files

#### Component Models

- Atmosphere Models
- Land Models
- Sea Ice Models
- Ocean Models
- Land Ice Models
- River Models
- Coupler

### EXTERNAL LIBRARY DOCUMENTATION

- Parallel I/O Library (PIO)
- Model Coupling Toolkit (MCT)
- Earth System Modeling Framework (ESMF)

### MODEL INPUT DATA

The input data necessary to run all supported component sets is made available from a public Subversion input data repository. Note that the input data repository has much more data in it than you need to run CESM2.0 — DO NOT attempt to svn checkout the whole input data repository. The CESM2.0 User's Guide explains how to obtain the subset of input data required for your needs.

#### PERFORMANCE AND LOAD BALANCING DATA

The development and testing of CESM2.0 required several iterations in the CESM1.Y series. The following timing tables for each model revision are available:

- CESM1.3 Timing Table
- CESM1.5 Timing Table
- CESM2.0 Timing Table

### **CESM PROJECT**

The Community Earth System Model (CESM) is a fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.

CESM is sponsored by the National Science Foundation (NSF) and the U.S. Department of Energy (DOE). Administration of the CESM is maintained by the Climate and Global Dynamics Laboratory (CGD) at the National Center for Atmospheric Research (NCAR).

### **MODEL SOURCE CODE**

### Copyright and Terms of Use

All CESM source code is subject to the following Copyright Notice and Disclaimer.

### Acquiring the Release Code

The source code for CESM release, is distributed through a public Subversion code repository. This code can be checked out using Subversion client software, such as the command tool swn, or simply view the latest version with a web

A short registration is required to access the repository. After registering, you will receive an email containing a user name and password that is necessary to gain access to the repository.

described in the code is more fully described in the most recent version of the CESM1.2 User's Guide.

### REPORTING A PROBLEM

If you have any problems, please first read the User's Guide including the sections on FAQs and Use Cases. Please also refer to the CESM. Bulletin Board, which is in place to facilitate communication within the CESM community. Finally, please also refer to the Release Notes entries that are provided with every release and release update. If questions or problems still exist, then please send an email to cesm-help@cgd.ucar.edu. Support questions will be answered as resources are available.

### CESM SUPPORT POLICY

### Right hand column has a link to the registration page, click on it



Register -- you will be emailed a username and password

# (C) Hardware/Software Requirements

### Supported platforms

**CESM** currently runs "out of the box" today on the following machines

- cheyenne NCAR IBM
- titan ORNL Cray XK6
- hopper NERSC Cray XE6
- edison NERSC Cray Cascade
- bluewaters ORNL Cray XE6
- intrepid ANL IBM Bluegene/P
- mira ANL IBM Bluegene/Q
- janus Univ Colorado HPC cluster
- pleiades NASA SGI ICE cluster
- and a few others



out of the box = works immediately after installation without any modification

## •Running CESM on other platforms Require porting + software

- Subversion client (version 1.4.2 or greater)
- Fortran and C compilers (recommend pgi, intel, or ibm xlf compilers)
- NetCDF library (recommend netcdf4.1.3 or later)
- MPI (MPI1 is adequate, Open MPI or MPICH seem to work on Linux clusters)

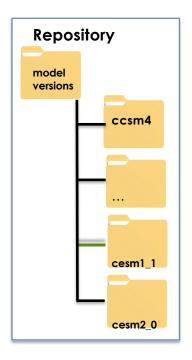
# (B) Download the Source Code

Code and input datasets are in a subversion repository (\*)

https://svn-ccsm-release.cgd.ucar.edu/model\_versions

List the versions available on the CESM repository

svn list https://svn-ccsm-release.cgd.ucar.edu/model\_versions



Check out a working copy from the repository ("Download code")

svn co https://svn-ccsm-release.cgd.ucar.edu/model\_versions/cesm2\_0

Note: only available upon CESM2.0 release, estimated December 2016

(\*) You can get subversion at <a href="http://subversion.apache.org/">http://subversion.apache.org/</a>

