CLM5.0 Tutorial: Running CLM

Danica Lombardozzi

NCAR is sponsored by the National Science Foundation

CLM Tutorial Practical Sessions: Week Overview

Today

Practical 1: Running CLM

- Log in to cheyenne computing environment
- Download CTSM code
- Run an out-of-the-box CLM simulation
- Basic visualization of model output

Tuesday

Practical 2: Changing Model Setup

- Changing component sets
- Basic namelist changes
- Parameter changes
- Simple data analysis

Wednesday

Practical 3: Changing Model Behavior

- Coding best practices
- Model behavior changes using namelists
- Code modifications
- Tracking down errors

Thursday

Practical 4: Single Point Simulations

- Set up and run single point simulations
- Considerations for model spin up
- Running transient historical simulations

Friday

Practical 5: FATES

- Set up and run a FATES simulation
- Analyzing FATES output

Outline

Lecture/Intro

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

Practical

- **Download CLM code** (this is a one time setup step)
- **Basic workflow**
  1) Create a new case
  2) Invoke case_setup
  3) Build the executable
  4) Run and output data
- **Finding & Looking at model output**

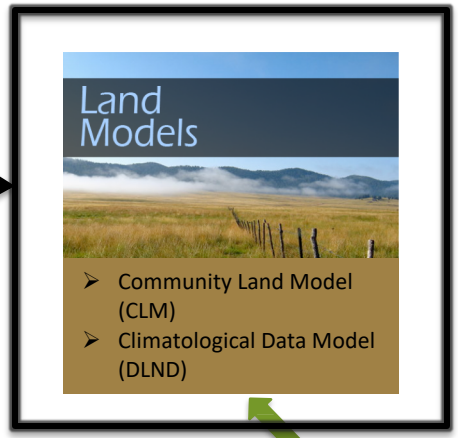
Other Useful Info

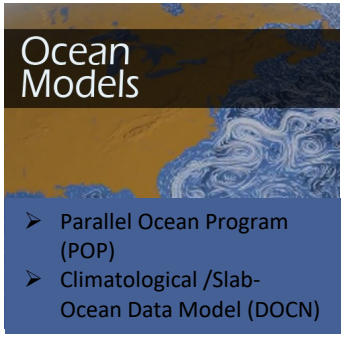
- **Getting help**

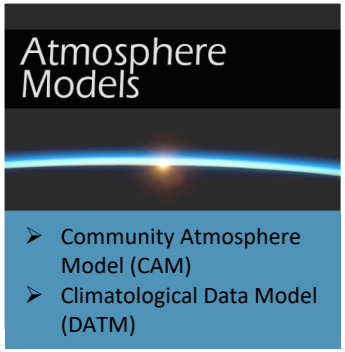
The CESM Framework

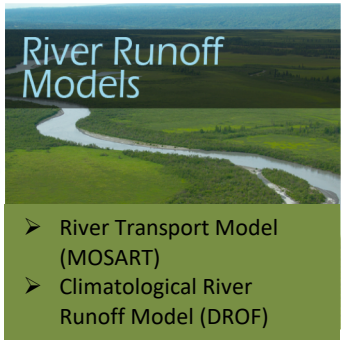
The Community Earth System Model (CESM) is a set of models that can be run independently or together to simulate the Earth global climate.

The CLM is the focus of this week's tutorial

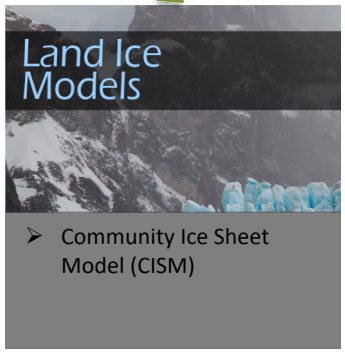








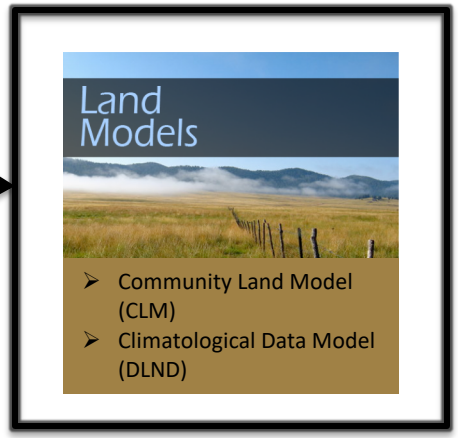
Coupler



The CESM Framework

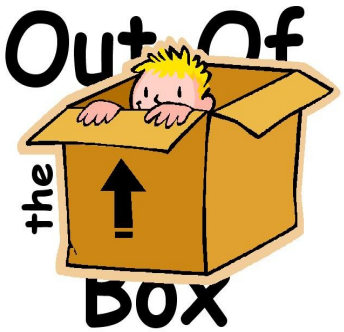
The Community Earth System Model (CESM) is a set of models that can be run independently or together to simulate the Earth global climate.

The CLM is the focus of this week's tutorial



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

This practical session is a quick start to the CLM workflow (out-of-the-box)



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

CESM 2.0 Web Page

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

Current Release

The current CESM supported release is CESM 2.1.0

Learn more | View Experiments | Download current release

About CESM2

CESM is a fully-coupled, community, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.

- What's New in CESM2
- CESM Naming Conventions
- Supported Releases and Known Issues

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.

- CESM2 Scientifically Validated Configurations
- Experiment Diagnostics
- Experiment Output Datasets *

** Please see NCAR Climate Data Gateway (formerly ESG) for data download details.*

CESM2 Quicklinks

Quick Start Guide
Downloading The Code
Scientifically Validated Configurations
Prognostic Components
CESM Software Engineering

Related Information

Data Management & Distribution Plan
Development Project Policies & Terms of Use
DiscussCESM Forums Bulletin Board
CESM2 Copyright
CESM Support Policy
CESM2 Included Packages Copyright

Quick Start

See the selected links below to help you quickly get started with CESM2

- Getting Help
- CESM2 Use Cases
- CESM2 Quick Start Guide
- Download the CESM2 Code

CIME Documentation

Common Infrastructure for Modeling the Earth contains the coupling infrastructure, support scripts, data models and utility libraries needed to create a single-executable coupled Earth System Model.

** CIME does not contain any prognostics components and is available in a stand-alone package that can be compiled and tested with just its data components.*

- CIME User Guide

Prognostic Components

Each model component page contains descriptions and documentation for active or prognostic models.

- Atmosphere
- Land
- Land Ice
- Ocean
- River Runoff
- Sea Ice
- Wave

Configurations and Grids

Component configurations include settings required for CIME enabled models; both prognostic and data model components. These settings include:

- Grid Resolutions
- Component Sets
- Component Configuration Settings

** Includes Fortran namelists and CASEROOT variable definitions*

Supported Machines & Performance Data

- Supported Machines and Compilers
- Timing, Performance and Load Balancing Data
- Running on a Medium-Sized Linux Cluster
- Verify a Machine Port

External Library Documentation

- Parallel I/O Library (PIO)
- Model Coupling Toolkit (MCT)
- Earth System Modeling Framework (ESMF)
- External Python Based Tools *

* Support for these tools is currently limited to **NCAR** machines only

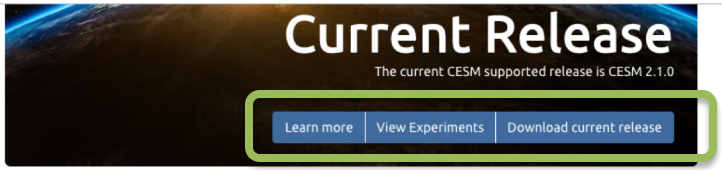
Model Input Data

As of CESM2, the input data necessary to run all supported component sets is made available from a number of different public repositories including:

GridFTP | Anonymous FTP | Subversion

CESM 2.0 Web Page

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



About CESM2

CESM is a fully-coupled, community, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.

- What's New in CESM2
- CESM Naming Conventions
- Supported Releases and Known Issues

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.

- CESM2 Scientifically Validated Configurations
- Experiment Diagnostics
- Experiment Output Datasets *

** Please see NCAR Climate Data Gateway (formerly ESG) for data download details.*

CESM2 Quicklinks

- Quick Start Guide
- Downloading The Code
- Scientifically Validated Configurations
- Prognostic Components
- CESM Software Engineering

Related Information

- Data Management & Distribution Plan
- Development Project Policies & Terms of Use
- DiscussCESM Forums Bulletin Board
- CESM2 Copyright
- CESM Support Policy
- CESM2 Included Packages Copyright

Quick Start

See the selected links below to help you quickly get started with CESM2

- Getting Help
- CESM2 Use Cases
- CESM2 Quick Start Guide
- Download the CESM2 Code

CIME Documentation

Common Infrastructure for Modeling the Earth contains the coupling infrastructure, support scripts, data models and utility libraries needed to create a single-executable coupled Earth System Model.

** CIME does not contain any prognostics components and is available in a stand-alone package that can be compiled and tested with just its data components.*

- CIME User Guide

Prognostic Components

Each model component page contains descriptions and documentation for active or prognostic models.

- Atmosphere
- Land
- Land Ice
- Ocean
- River Runoff
- Sea Ice
- Wave

Configurations and Grids

Component configurations include settings required for CIME enabled models; both prognostic and data model components. These settings include:

- Grid Resolutions
- Component Sets
- Component Configuration Settings

** Includes Fortran namelists and CASEROOT variable definitions*

Supported Machines & Performance Data

- Supported Machines and Compilers
- Timing, Performance and Load Balancing Data
- Running on a Medium-Sized Linux Cluster
- Verify a Machine Port

External Library Documentation

- Parallel I/O Library (PIO)
- Model Coupling Toolkit (MCT)
- Earth System Modeling Framework (ESMF)
- External Python Based Tools *

* Support for these tools is currently limited to **NCAR** machines only

Model Input Data

As of CESM2, the input data necessary to run all supported component sets is made available from a number of different public repositories including:

GridFTP | Anonymous FTP | Subversion

CLM Web Page

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

CLM5 Documentation

Introduction

CLM5.0 is the latest in a series of land models developed through the CESM project. More information on the CLM project and access to previous released CLM model versions and documentation can be found via the CLM Web Page. Note that CLM4.5 biogeophysics and biogeochemistry can be run from this release code. A new river model (MOSART) is also included. *This release is a land-only release.* The capability to run CLM5.0 within CESM2.0 will be included in the CESM2.0 release.

The Functionally Assembled Terrestrial Ecosystem Simulator [FATES] is available within the CLM5 release as a research option.

Access

- CLM5.0 is publicly available through the Community Terrestrial System Model (CTSM) git repository
- Download the code by executing the following commands:

```
$ git clone -b release-clm5.0 https://github.com/ESCOMP/ctsm.git clm5.0
$ cd clm5.0
$ ./manage_externals/checkout_externals
```

Documentation

- CLM5.0 Technical Description [HTML] [PDF]
- What's new in CLM5.0 Description [Text] [Image]
- CLM5.0 User's Guide
- CESM2.0 Quickstart Guide *Note that same script commands used for CLM land-only
- CIME documentation *CIME, pronounced "SEAM", is the Common Infrastructure for Modeling the Earth provides a UNIX command-line-based interface for configuring, compiling and executing Earth system models including CLM
- CLM namelist definitions
- CLM CASEROOT variable definitions
- MOSART namelist definitions
- Known bugs/issues in CLM
- Explanation of supported configurations in CLM5
- Getting Help - DiscussCESM Forums

Previous Versions

- CLM4.5 Technical Description [PDF]
- What's new in CLM4.5 [HTML]
- CLM4.0 Technical Description [PDF]
- CLM4.0 Urban Model Technical Description [PDF]
- What's new in CLM4.0 [HTML]

Tools

- Toolbox for Human-Earth System Integration and Scaling (THESIS)

Tutorials

- CLM/CTSM Tutorial Announcement (2019)
- FATES Tutorial (Feb, 2018)
- CLM Tutorial (Sept, 2016)

Model Design and Development

All future CLM development will occur within the framework of **CTSM**. CLM will be an instantiation of CTSM.

- Development Guide
  CTSM development guidelines, workflow, and coding standards provided at CTSM github wiki page

Model output and diagnostic plots

- CLM5.0, CLM4.5, and CLM4.0 land-only control simulations available on Cheyenne at the path below or inquire with Keith Oleson [oleson@ucar.edu]
  *PFT-level, daily, and hourly data available for selected fields and simulations)

```
/glade/p/cgd/tss/people/oleson/CLM_LAND_ONLY_RELEASE
```

- CLM Diagnostic package plots
- ILAMB package plots
- CESM Postprocessing Tool: Quick Start, User's Guide, and Workflow

CESM Project

CESM is a fully-coupled, community, 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).

CESM2 Quicklinks

Quick Start Guide
Downloading The Code
Scientifically Validated Configurations
› Prognostic Components
CESM Software Engineering

Related Information

Data Management & Distribution Plan
Development Project Policies & Terms of Use
DiscussCESM Forums Bulletin Board
CESM2 Copyright
CESM Support Policy
CESM2 Included Packages Copyright

CLM Web Page

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

CLM5 Documentation

Introduction

CLM5.0 is the latest in a series of land models developed through the CESM project. More information on the CLM project and access to previous released CLM model versions and documentation can be found via the CLM Web Page. Note that CLM4.5 biogeophysics and biogeochemistry can be run from this release code. A new river model (MOSART) is also included. *This release is a land-only release.* The capability to run CLM5.0 within CESM2.0 will be included in the CESM2.0 release.

The Functionally Assembled Terrestrial Ecosystem Simulator [FATES] is available within the CLM5 release as a research option.

Access

- CLM5.0 is publicly available through the Community Terrestrial System Model (CTSM) git repository
- Download the code by executing the following commands:

```
$ git clone -b release-clm5.0 https://github.com/ESCOMP/ctsm.git clm5.0
$ cd clm5.0
$ ./manage_externals/checkout_externals
```

Documentation

- CLM5.0 Technical Description [HTML] [PDF]
- What's new in CLM5.0 Description [Text] [Image]
- CLM5.0 User's Guide
- CESM2.0 Quickstart Guide *Note that same script commands used for CLM land-only
- CIME documentation *CIME, pronounced "SEAM", is the Common Infrastructure for Modeling the Earth provides a UNIX command-line-based interface for configuring, compiling and executing Earth system models including CLM
- CLM namelist definitions
- CLM CASEROOT variable definitions
- MOSART namelist definitions
- Known bugs/issues in CLM
- Explanation of supported configurations in CLM5
- Getting Help - DiscussCESM Forums

Previous Versions

- CLM4.5 Technical Description [PDF]
- What's new in CLM4.5 [HTML]
- CLM4.0 Technical Description [PDF]
- CLM4.0 Urban Model Technical Description [PDF]
- What's new in CLM4.0 [HTML]

Tools

- Toolbox for Human-Earth System Integration and Scaling (THESIS)

Tutorials

- CLM/CTSM Tutorial Announcement (2019)
- FATES Tutorial (Feb, 2018)
- CLM Tutorial (Sept, 2016)

Model Design and Development

All future CLM development will occur within the framework of **CTSM**. CLM will be an instantiation of CTSM.

- Development Guide
  CTSM development guidelines, workflow, and coding standards provided at CTSM github wiki page

Model output and diagnostic plots

- CLM5.0, CLM4.5, and CLM4.0 land-only control simulations available on Cheyenne at the path below or inquire with Keith Oleson [oleson@ucar.edu]
  *PFT-level, daily, and hourly data available for selected fields and simulations)

```
/glade/p/cgd/tss/people/oleson/CLM_LAND_ONLY_RELEASE
```

- CLM Diagnostic package plots
- ILAMB package plots
- CESM Postprocessing Tool: Quick Start, User's Guide, and Workflow

CESM Project

CESM is a fully-coupled, community, 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).

CESM2 Quicklinks

Quick Start Guide
Downloading The Code
Scientifically Validated Configurations
› Prognostic Components
CESM Software Engineering

Related Information

Data Management & Distribution Plan
Development Project Policies & Terms of Use
DiscussCESM Forums Bulletin Board
CESM2 Copyright
CESM Support Policy
CESM2 Included Packages Copyright

CLM Web Page

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

CLM5 Documentation

Introduction

CLM5.0 is the latest in a series of land models developed through the CESM project. More information on the CLM project and access to previous released CLM model versions and documentation can be found via the CLM Web Page. Note that CLM4.5 biogeophysics and biogeochemistry can be run from this release code. A new river model (MOSART) is also included. *This release is a land-only release.* The capability to run CLM5.0 within CESM2.0 will be included in the CESM2.0 release.

The Functionally Assembled Terrestrial Ecosystem Simulator [FATES] is available within the CLM5 release as a research option.

Access

- CLM5.0 is publicly available through the Community Terrestrial System Model (CTSM) git repository
- Download the code by executing the following commands:

```
$ git clone -b release-clm5.0 https://github.com/ESCOMP/ctsm.git clm5.0
$ cd clm5.0
$ ./manage_externals/checkout_externals
```

Documentation

- CLM5.0 Technical Description [HTML] [PDF]
- What's new in CLM5.0 Description [Text] [Image]
- CLM5.0 User's Guide
- CESM2.0 Quickstart Guide *Note that same script commands used for CLM land-only
- CIME documentation *CIME, pronounced "SEAM", is the Common Infrastructure for Modeling the Earth provides a UNIX command-line-based interface for configuring, compiling and executing Earth system models including CLM
- CLM namelist definitions
- CLM CASEROOT variable definitions
- MOSART namelist definitions
- Known bugs/issues in CLM
- Explanation of supported configurations in CLM5
- Getting Help - DiscussCESM Forums

Previous Versions

- CLM4.5 Technical Description [PDF]
- What's new in CLM4.5 [HTML]
- CLM4.0 Technical Description [PDF]
- CLM4.0 Urban Model Technical Description [PDF]
- What's new in CLM4.0 [HTML]

Tools

- Toolbox for Human-Earth System Integration and Scaling (THESIS)

Tutorials

- CLM/CTSM Tutorial Announcement (2019)
- FATES Tutorial (Feb, 2018)
- CLM Tutorial (Sept, 2016)

Model Design and Development

All future CLM development will occur within the framework of **CTSM**. CLM will be an instantiation of CTSM.

- Development Guide
  CTSM development guidelines, workflow, and coding standards provided at CTSM github wiki page

Model output and diagnostic plots

- CLM5.0, CLM4.5, and CLM4.0 land-only control simulations available on Cheyenne at the path below or inquire with Keith Oleson [oleson@ucar.edu]
  *PFT-level, daily, and hourly data available for selected fields and simulations)

```
/glade/p/cgd/tss/people/oleson/CLM_LAND_ONLY_RELEASE
```

- CLM Diagnostic package plots
- ILAMB package plots
- CESM Postprocessing Tool: Quick Start, User's Guide, and Workflow

CESM Project

CESM is a fully-coupled, community, 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).

CESM2 Quicklinks

Quick Start Guide

Downloading The Code

Scientifically Validated Configurations

› Prognostic Components

CESM Software Engineering

Related Information

Data Management & Distribution Plan

Development Project Policies & Terms of Use

DiscussCESM Forums Bulletin Board

CESM2 Copyright

CESM Support Policy

CESM2 Included Packages Copyright

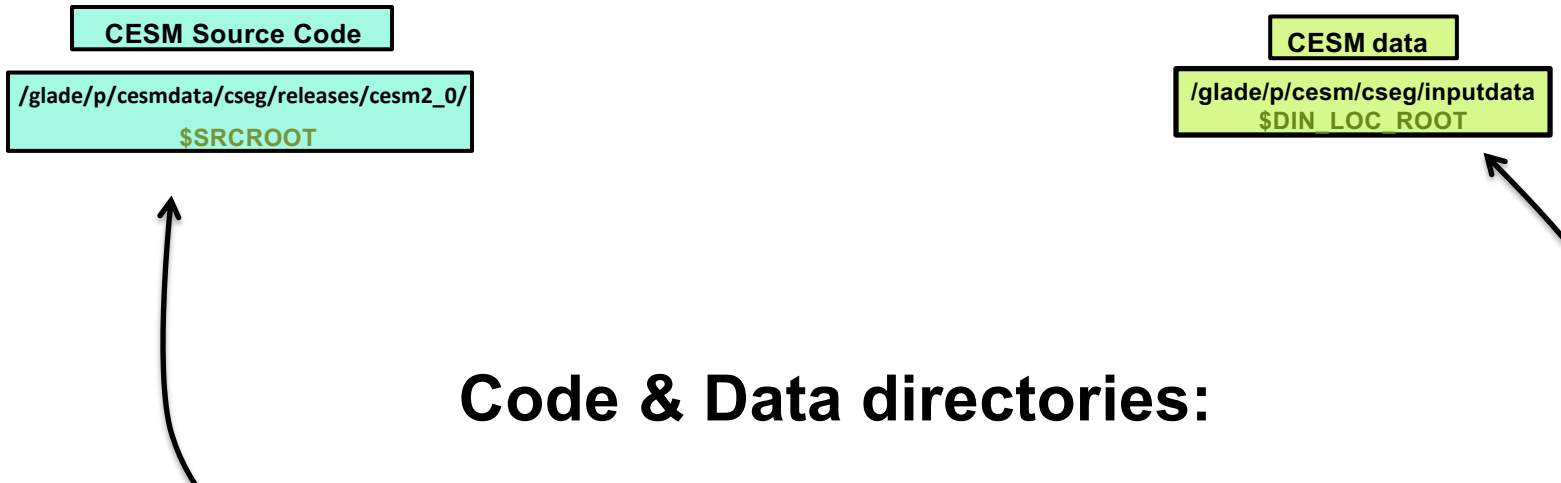
	CLM4				CLM4.5				CLM5			
Forcing	SP	BGC	+N, +CO_2	no LULCC	SP	BGC	+N, +CO_2	no LULCC	SP	BGC crop	+N, +CO_2	no LULCC
GSWP3 v1	✓^{o}	✓^{o} ✸	✓	✓	✓	✓^{o} ✸	✓	✓	✓^{o}	✓^{o} ✸	✓	✓
CRUNCEP v7		✓				✓			✓	✓ ✸		✓
WATCH/ WFDEI									✓^{WF}	✓^{W}		

✓ Historical simulation (1850-2014, ^{W} 1850-2001, ^{WF} 1979-2014)

✸ Projection period simulations (RCP8.5 2015-2300)

^{o} Daily and hourly output

CESM Directory Structure



Code & Data directories:

- **CESM source code**
  /glade/p/cesmdata/cseg/releases/cesm2_0/

***Note**: This is the released version of the code on Cheyenne. You can use this code if you do not plan to make any code changes. For this tutorial, we will download a copy of the code to your home directory and eventually make code changes, so the source code path will be different.*

- **CESM data**
  /glade/p/cesm/cseg/inputdata

***Note**: The data used by the model live here.*

CESM Directory Structure

CESM Source Code

/glade/p/cesmdata/cseg/releases/cesm2_0/

$SRCROOT

- **components**
  - **cam (atmosphere)**
  - **cice (sea ice)**
  - **cism (land ice)**
  - **clm (land)**
  - **mosart (river routing)**
  - **pop (ocean)**
  - **rtm (river routing)**
  - **ww3 (ocean waves)**
- **cime**
  - **scripts**
    create_newcase

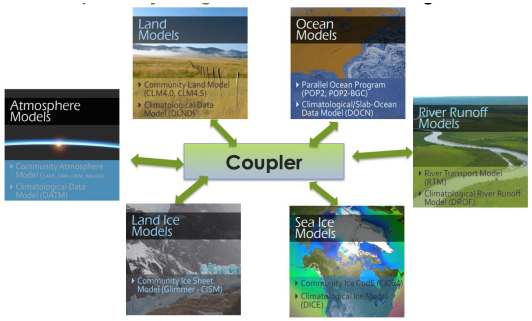
CESM data

/glade/p/cesm/cseg/inputdata

$DIN_LOC_ROOT

Source code has 2 important subdirectories:

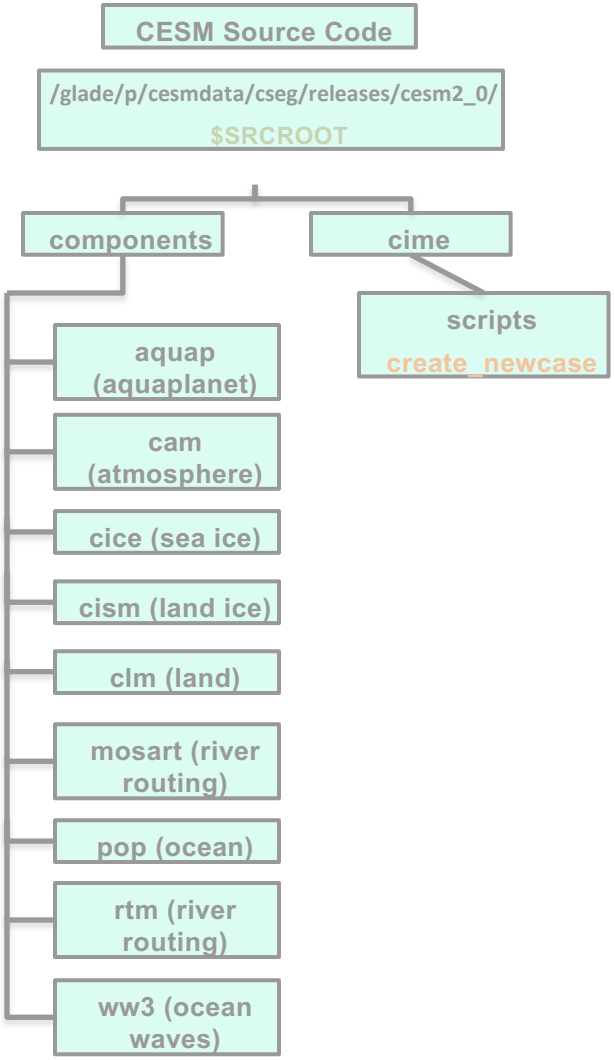
- components: contains the code for every model component

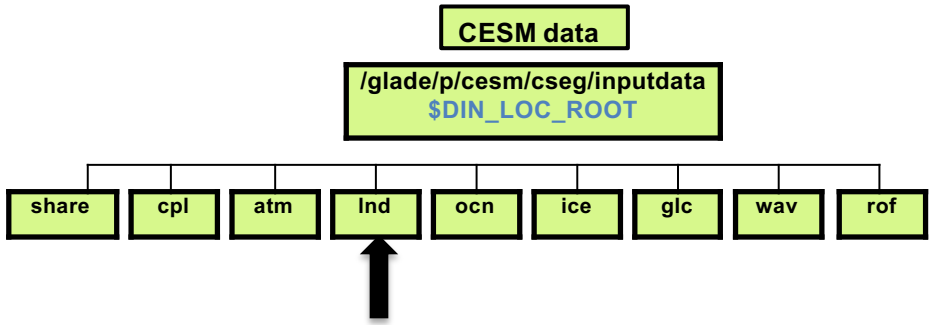


- cime: contains the scripts you need to run CESM

Note: the subdirectories of "components" will change based on whether you are using a CESM code base (shown here) or a CLM code base. For CLM-only code base, you will only find 'mosart' , 'rtm', and 'cism' subdirectories, and code for CLM in the 'src' subdirectory. 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 later.

CESM Directory Structure





"Inputdata" directory, $DIN_LOC_ROOT, contains all input data required to run the model

Note: There are many subdirectories, including one for each component of CESM.

Outline

Lecture/Intro

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

Practical

- **Download CLM code** (this is a one time setup step)
- **Basic workflow**
  1) Create a new case
  2) Invoke case_setup
  3) Build the executable
  4) Run and output data
- **Finding & Looking at model output**

Other Useful Info

- **Getting help**

Work Flow: Super Quick Start

CLM 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

In this session, you will learn to use these four steps to set up and run a simulation.

Start Practical Here

First: Logging in to Cheyenne

1. **Open a secure shell window on your computer**:
   Example programs: Terminal, Cygwin, PuTTY, Mobaxterm

2. **Log on using either Duo or your yubikey:**
   `ssh -Y <username>@cheyenne.ucar.edu`

Note: Throughout the tutorial, action steps like this are identified by the green text. Places where you need to change text will be bracketed with "< >"

Your screen displays a response:
Token_response:

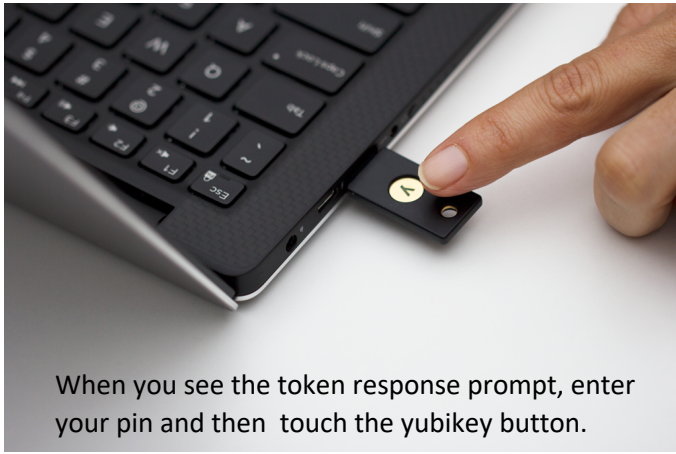
To access cheyenne, you can use either Duo Mobile authentication or a Yubikey authentication

First: Logging in to Cheyenne

If you are using a Yubikey

3. **Enter your PIN number (do not hit enter), then touch the yubikey button.** This will insert a new one-time password and a return

Note: the yubikey is activated by the warmth of your finger, not the pressure of pushing the button



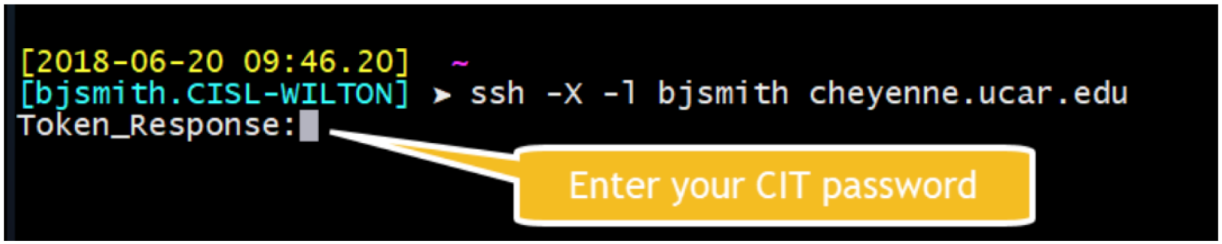
When you see the token response prompt, enter your pin and then touch the yubikey button.

First: Logging in to Cheyenne

If you are using a Duo Mobile

To log in to a system like Cheyenne:

- Enter your ssh command.
- Enter your **CIT password** where a token response is requested.



The Duo App will send a request (a "push" notification) to your phone or tablet, asking you to approve or deny the login request.

When you approve the request, you will be logged in.

Download the app here:
https://duo.com/product/trusted-users/two-factor-authentication/duo-mobile

One time setup: Download CLM code

Note: CLM5.0 is publicly available through the Community Terrestrial System Model (CTSM) git repository (https://github.com/ESCOMP/ctsm.git). The steps below will show you how to download the CLM5.0 code and are available on the CLM website. Downloading the full CESM code is similar, but requires a different git repository. To download the full CESM code, follow the quick start guide on the CESM webpage

To Do:

1) Navigate to your home directory

```
cd ~
```

Note: The character " ~ " is a shortcut to your home directory. On cheyenne, your home directory is /glade/u/home/<username>

2) Download the CLM5.0 code to your home directory

```
git clone -b release-clm5.0 https://github.com/ESCOMP/ctsm.git clm5.0_2019tutorial
```

3) Navigate to the code directory you just downloaded

```
cd clm5.0_2019tutorial
```

4) Checkout all the model components

```
./manage_externals/checkout_externals
```

Outline

Lecture/Intro

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

Practical

- **Download CLM code** (this is a one time setup step)
- **Basic workflow**
  1) Create a new case
  2) Invoke case_setup
  3) Build the executable
  4) Run and output data
- **Finding & Looking at model output**

Other Useful Info

- **Getting help**

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

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

(1) create a new case

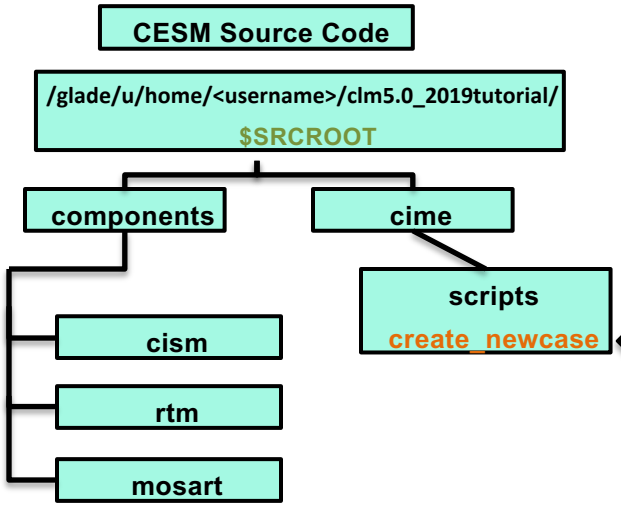
(2) invoke case.setup

(3) build the executable

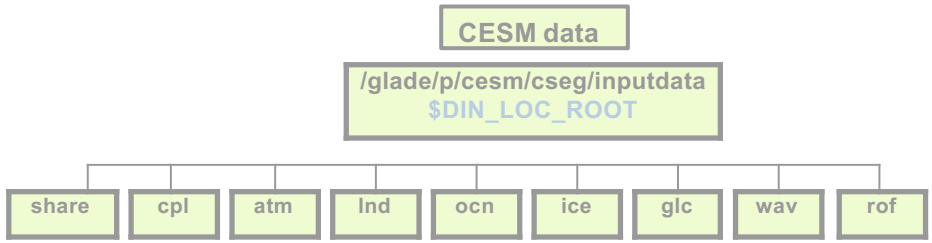
(4) submit your run to the batch queue

We will progress step by step, starting with step 1

CLM Directory Structure



the **script** you need **to set up a new simulation** (or **create a new case**)



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

Follow the steps on the next slide to set up a simulation

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

Note that "$USER" is an alias for your username.

Note: If you don't have a .cesmproj file set up in your home directory, you need to specify an account in create_newcase

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

Note that "./" is how you tell the computer to execute a script

Stop here

---

(2) invoke case.setup

(3) build the executable

(4) submit your run to the batch queue

Next, let's dig into the details of this command to understand the parts

Create a new case

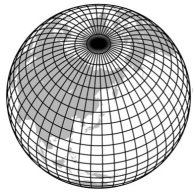
In the scripts directory, create_newcase is the tool that generates a new case.

create_newcase requires 3 arguments

What is the casename ?

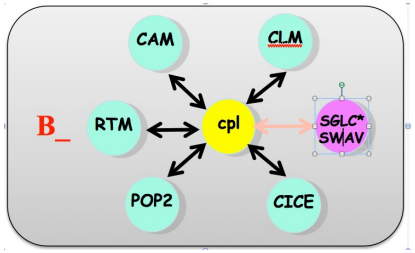
-- case

Which resolution?



-- res

Which model configuration and components ?



-- compset

***Note**: A 4th argument, "--mach", is required if you are running on an unsupported machine. It is no longer required on supported machines*

Creating a new case

In the scripts directory, create_newcase is the tool that generates a new case.

create_newcase requires 3 arguments

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp
```

case specifies location and name of the case being created

~/clm_tutorial_cases/I1850CLM50_001

"~" = home directory, or /glade/u/home/<username>
"/clm_tutorial_cases" = the subdirectory we created to store your cases
"I1850CLM50_001" = case directory name

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

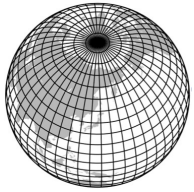
Note: Steps 2-4 take place in the case directory that you create here in step 1. More on that coming up.

Creating a new case

In the scripts directory, create_newcase is the tool that generates a new case.

create_newcase requires 3 arguments

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp
```



res specifies the model resolutions (or grid): f19_g17 is a 2-degree grid

Grid naming convention

Each model resolution can be specified by its alias or long name.

Example of equivalent alias and long name:

- **alias: f19_g17 (atm/lnd_ocn/ice)**
- **long name = a%1.9x2.5_l%1.9x2.5_oi%gx1v7_r%r05_g%gland4_w%ww3a_m%gx1v7**

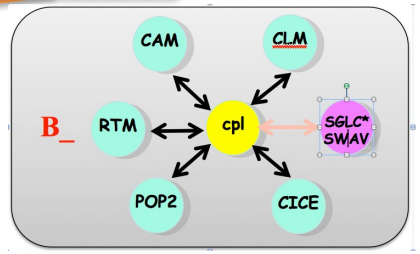
atm	lnd	ocn/ice grid	river	lnd-ice	wave	ocn-ice mask

Creating a new case

In the scripts directory, create_newcase is the tool that generates a new case.

create_newcase requires 3 arguments

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp
```



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 or long name. Example of an equivalent alias and long name:

- **alias: I1850Clm50Sp**
- **long name =** 1850_DATM%GSWP3v1_CLM50%SP_SICE_SOCN_MOSART_CISM2%NOEVOLVE_SWAV

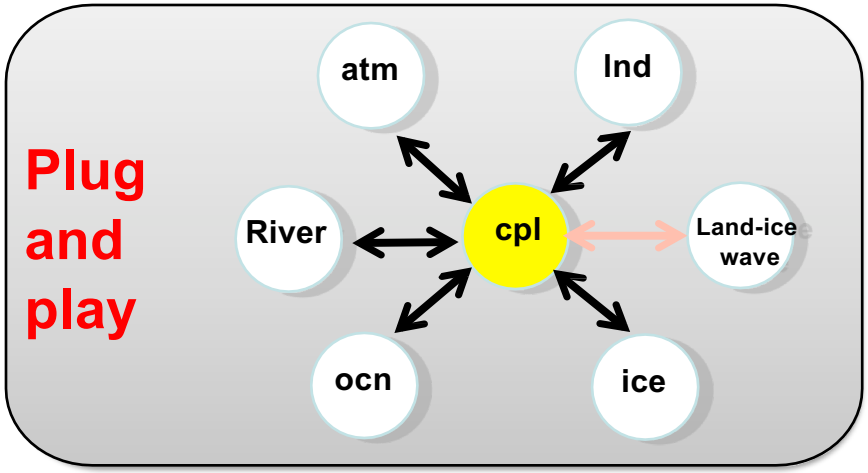
time atm lnd ice ocn river lnd-ice wave

Note: Some compsets are scientifically supported and others are not. You can use an unsupported compset, but will need to add the option "--run_unsupported" at the end of the create_newcase command line

More on CESM component sets

Plug and play of components with different component models

Color code: active data stub



Key Definitions:

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

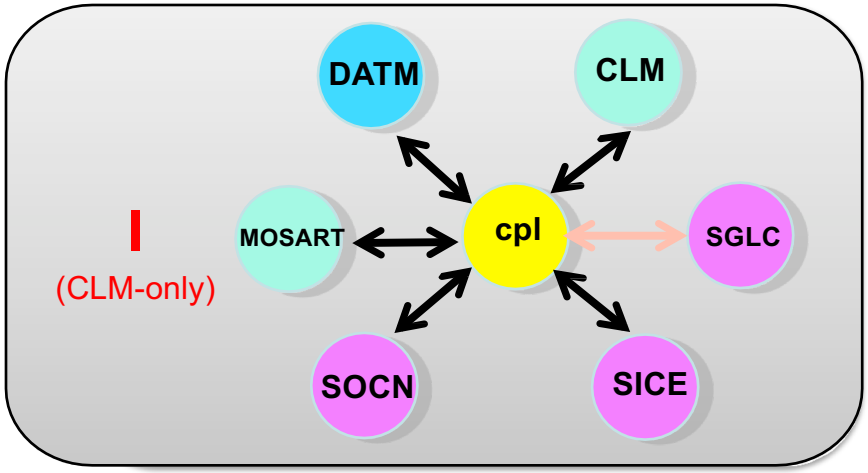
Data: Simulation is reading in data from a file for this component

Stub: Component is not being used

More on CESM component sets

Plug and play of components with different component models

Color code: active data stub



Key Definitions:

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

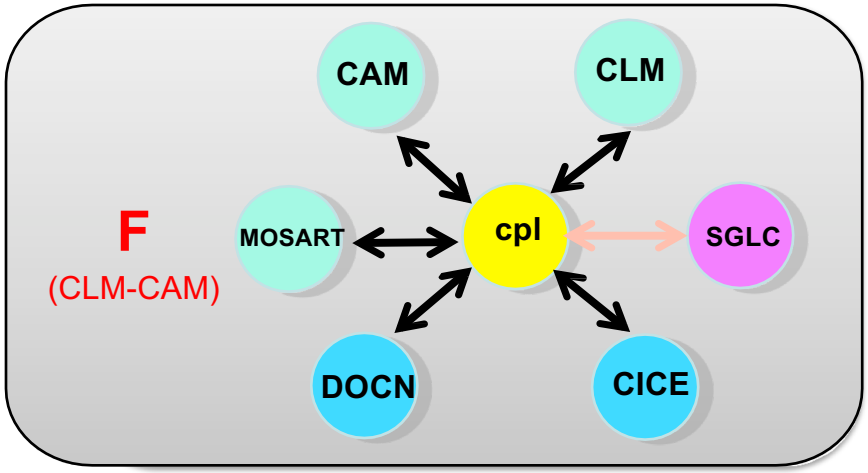
Data: Simulation is reading in data from a file for this component

Stub: Component is not being used

More on CESM component sets

Plug and play of components with different component models

Color code: active data 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 available compsets, grids, etc.:

```
./query_config --h
```

This will show a help message with information and options for the command

For example*:*

```
./query_config --compsets clm
```

Will list all the "I" compsets available

```
./query_config --grids
```

Will list all the available model grids

You'll explore different compsets in Practical 2

CESM 2.0 Web Page

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

List of compsets and grids are also available on the CESM website

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

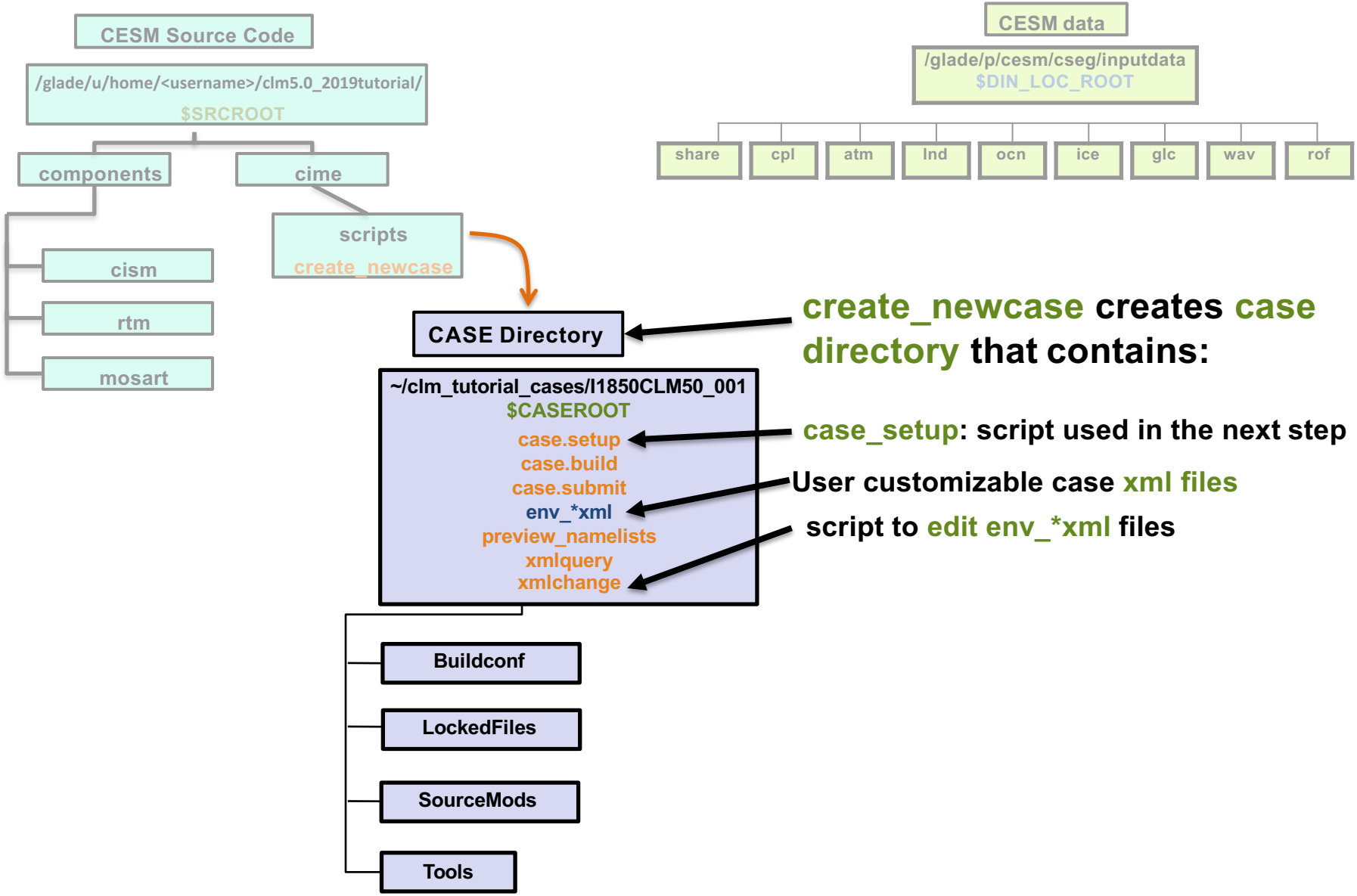
This command line creates a case directory with the case name you specified. Next, let's explore that directory structure.

(2) invoke case.setup

(3) build the executable

(4) submit your run to the batch queue

CLM Directory Structure



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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

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: Create & run an out-of-the-box simulation

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(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

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

Start here

(2) invoke case.setup

Navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

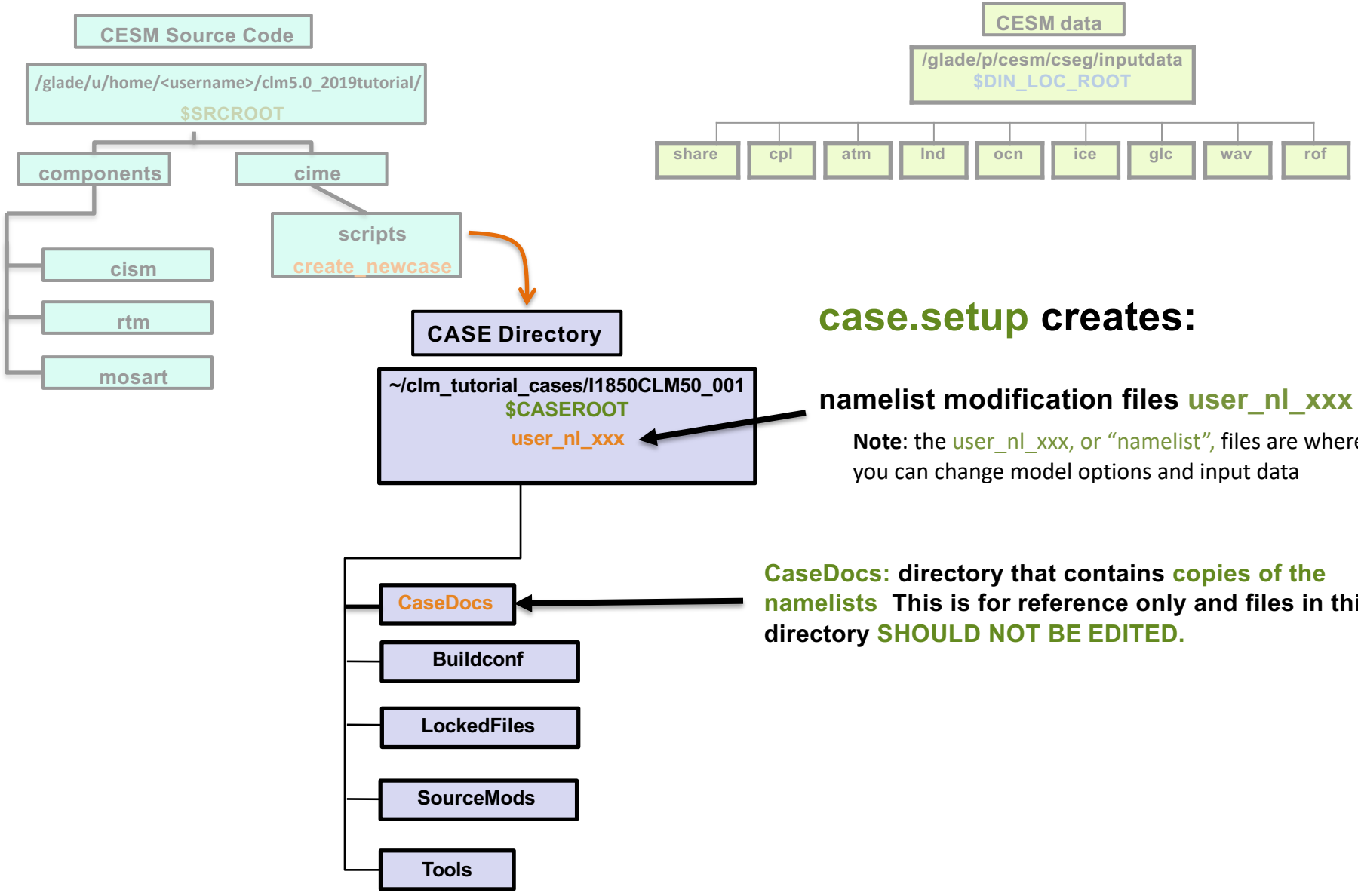
```
./case.setup
```

Stop here

(3) build the executable

(4) submit your run to the batch queue

CLM Directory Structure



case.setup creates:

namelist modification files user_nl_xxx

Note: the user_nl_xxx, or "namelist", files are where you can change model options and input data

CaseDocs: directory that contains copies of the namelists This is for reference only and files in this directory SHOULD NOT BE EDITED.

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./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: Create & run an out-of-the-box simulation

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Then, navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./case.setup
```

(3) build the executable

Next, we will compile the model code

(4) submit your run to the batch queue

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Then, navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./case.setup
```

***If you had an account on cheyenne before the tutorial**, make sure the PBS_ACCOUNT is set to UCGD0004 before you build.*

Start here

(3) build the executable

Type this command line:

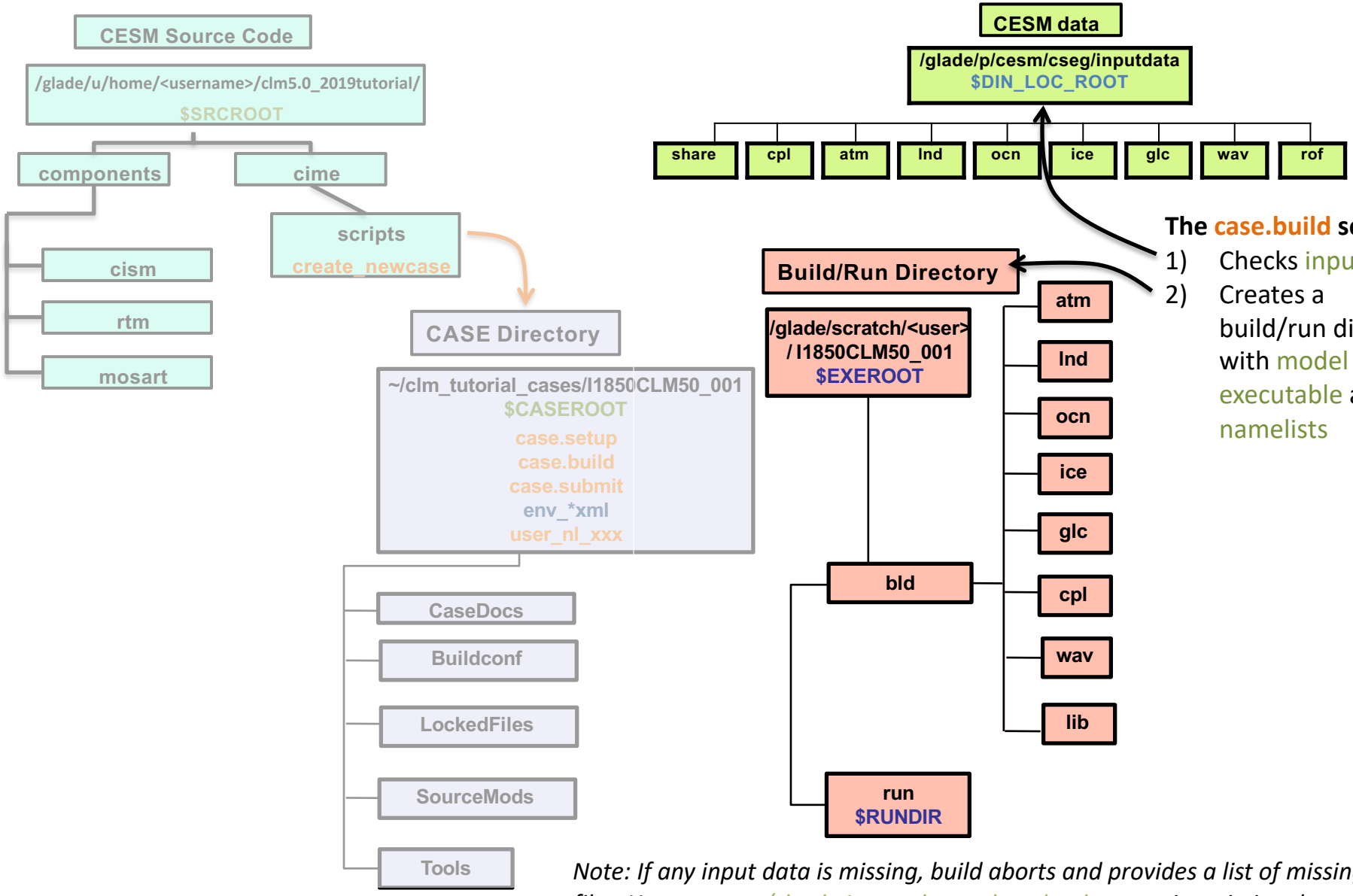
```
qcmd -q R4230874 -- ./case.build
```

Note that "qcmd --" is specific for cheyenne and runs the command on a computing node, reducing the load on the login node. Also, "-q" is specific for the tutorial and allows us to use special reserved nodes. You don't normally need to include the "-q", but must include "qcmd --" when running on cheyenne.

Stop here

(4) submit your run to the batch queue

CLM Directory Structure



The case.build script:

1) Checks input data
2) Creates a build/run directory with model executable and namelists

Note: If any input data is missing, build aborts and provides a list of missing files. You can run ./check_input_data --download to acquire missing data

A note on queue reservations and project code for the tutorial

For this tutorial, we have a special job queue reserved. Normally, your job will default to the 'regular' queue (you can also change to economy or premium), but this week we will use the reservation numbers so that we can access the specially reserved nodes. Below is a table of this weeks queue reservations

Date	Queue
Monday, Feb. 4	R4230874
Tuesday, Feb. 5	R4231039
Wednesday, Feb. 6	R4231261
Thursday, Feb. 7	R4231266
Friday, Feb. 8	R4231271

Also note that the tutorial uses project code **UCGD0004**

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Then, navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./case.setup
```

(3) build the executable

Type this command line:

```
qcmd -q R4230874 -- ./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

Case Customization Checks

The model is now compiled and ready to run! There are a few things we should check before submitting the run. For example:

1) How many days or years will the model simulate?
2) How much time does the computer need for this simulation?
3) Which computing project account is the model charging to?

These options are specified in the "env_*.xml" files in your case directory

Next, we'll review how to check and modify variables in XML files

The XML files can be modified directly, but we recommend that you use the xmlchange script.

How To: Changing case options using xmlchange

Using the "xmlchange" script is the preferred method, but you can edit XML files by hand

Benefits of using the "xmlchange" script:

1. Allows changing variables in env_*.xml files using a command-line interface
2. Won't let you mess up the syntax! The script checks the setting immediately for validity.
3. Settings are copied into the CaseStatus file, providing documentation of your changes.

env_*.xml file descriptions

Here is a list of the XML files in your case directory and a description of they are

File Name	Description
env_archive.xml	Specifies rules for short-term archival script case.st_archive
env_batch.xml	Set by create_newcase to define batch specific settings used by the script case.submit, including project number and computing time
env_build.xml	Specifies build information used by the script case.build. Note that if this is modified, the model must be recompiled
env_case.xml	Set by create_newcase and cannot be modified
env_mach_pes.xml	Specifies the PE layout of components used by the script case.run
env_mach_specific.xml	Specifies machine-specific information used by the script case.build
env_run.xml	Sets run-time information such as the length of the run, frequency of restarts. This is the most frequently modified xml file.

env_*.xml file descriptions

Here is a list of the XML files in your case directory and a description of they are

File Name	Description
env_archive.xml	Specifies rules for short-term archival script case.st_archive
env_batch.xml	Set by create_newcase to define batch specific settings used by the script case.submit, including project number and computing time
env_build.xml	Specifies build information used by the script case.build. Note that if this is modified, the model must be recompiled
env_case.xml	Set by create_newcase and cannot be modified
env_mach_pes.xml	Specifies the PE layout of components used by the script case.run
env_mach_specific.xml	Specifies machine-specific information used by the script case.build
env_run.xml	Sets run-time information such as the length of the run, frequency of restarts. This is the most frequently modified xml file.

The env_batch.xml and env_run.xml files include most of the variables you will need to modify to set up and run simulations and can be changed at any time before running the simulation.

How To: Changing case options using xmlchange

A few useful tips for using the xml scripts:

1. Use "*./xmlquery --listall*" to list variables and their values in the .xml files
2. Modify a variable in a .xml file, use *"./xmlchange"*
3. For help, type *./xmlchange --help*

Example: editing env_*.xml via the xmlchange tool

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

Next, let's modify a few important variables in the XML files

Many runtime variables are found in the env_run.xml file. The variables in this file control the mechanics of the run (length, resubmits, and archiving).

Common variables in env_run.xml to change include:

1. STOP_OPTION → sets the run-time interval type, i.e. nmonths, ndays, nyears

2. STOP_N→ sets the number of run-time 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

To Do: *Use xmlquery to find the values of the variables listed above. Feel free to play around with options*

```
./xmlquery STOP_OPTION
```

Note: Use `./xmlquery --listall` to find the values of all variables in all XML files, or `./xmlquery --file <env_*.xml>` to list variables in a specific file. You can also search for multiple variables, separating the variable names with a comma (","). You can also search for strings using `./xmlquery -p <string>`. This will search for every variable that includes this string. Try it for "STOP" or "CLM".

You should find these values:

1. **STOP_OPTION** → ndays
2. **STOP_N** → 5
3. **RESUBMIT** → 0

By default, the model is set to run for 5 days. Let's change the length of the simulation to 5 years.*

**We will not see any model output from a 5-day model run because history files with model output are only recorded every month by default*

To Do:

```
./xmlchange STOP_OPTION=nyears
```

This changes the run-time interval from days to years. Verify that the change worked as you expected:

```
./xmlquery STOP_OPTION
```

Now that you have confirmed that the run-time interval is years, verify that the model will run for 5 years:

```
./xmlquery STOP_N
```

Note: If you try to change a variable to a value that isn't an option, you will get an error message with a list of valid values.

We also need to check that we are using the correct project code and giving the computer enough time to run the simulation. These are found in the env_batch.xml file.

To Do: Use xmlquery to check

1. **PROJECT**

2. **JOB_WALLCLOCK_TIME**

Do you remember how to do this? If not, review the command-line prompts on previous two slides.

Did you find the following?

1. PROJECT→ UCGD0004

This is the project number we will use for the tutorial. If your project number is different from this, please use the xmlchange command to update to this project number

2. JOB_WALLCLOCK_TIME → 12:00:00

It won't take 12 hours to run a 5-year simulation and your simulation will get into the queue more quickly if you set a shorter run time.

A typical CLM-only 1850 satellite phenology (SP) simulation will run ~115 years in 12 hours.

How long will it take to run 5 years?

(I usually round **up** to the nearest 30 minute interval)

To Do: Change the JOB_WALLCLOCK_TIME to the amount of time you estimate your simulation will take (the answer to the above question)

Do you remember how to do this? If not, review the command-line prompts on previous slides.

Note: You can find out timing information for some standard simulations here: https://csegweb.cgd.ucar.edu/timing/cgi-bin/timings.cgi. Click on a compset similar to the one you will run and look for the "model throughput" value. This is how many simulated years the computer can run in a 24-hour timeframe, and you can estimate the amount of time you'll need based on this.

Case Customization Checks

To review, the variables associated with these questions are:

1) How many days or years will the model simulate?
   STOP_OPTION, STOP_N

2) How much time does the computer need for this simulation?
   JOB_WALLCLOCK_TIME

3) Which computing project account is the model charging to?
   PROJECT

There may be other variables you'll eventually want to change, and you now know how to use these tools to do that.

Now that we've customized the simulation run-time options, we just need to do the last step: submit the simulation! Let's first review what we've done so far

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Then, navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./case.setup
```

(3) build the executable

Type this command line:

```
qcmd -q R4230874 -- ./case.build
```

Change run-time options using the xmlquery and xmlchange command

(4) submit your run to the batch queue

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

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Then, navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./case.setup
```

(3) build the executable

Type this command line:

```
qcmd -q R4230874 -- ./case.build
```

Start here

(4) submit your run to the batch queue

Type this command line:

```
./case.submit
```

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

To Do:

First, a one-time step to create a directory to store your experiment cases:

```
mkdir ~/clm_tutorial_cases
```

Then, navigate to the scripts directory in the source code directory:

```
cd /glade/u/home/$USER/clm5.0_2019tutorial/cime/scripts
```

(1) create a new case

Type this command line:

```
./create_newcase --case ~/clm_tutorial_cases/I1850CLM50_001 --res f19_g17 --compset I1850Clm50Sp --project UCGD0004
```

(2) invoke case.setup

Then, navigate to your case directory:

```
cd ~/clm_tutorial_cases/I1850CLM50_001
```

Type this command line:

```
./case.setup
```

(3) build the executable

Type this command line:

```
qcmd -q R4230874 -- ./case.build
```

Start here

(4) submit your run to the batch queue

Type this command line:

```
./case.submit
```

Congratulations on submitting your simulation! Next, let's check the confirmation message & job status

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
Submit job case.run
Submitting job script qsub -q regular -l walltime=12:00:00 -A P93300641 -v ARGS_FOR_SCRIPT='--resubmit' .case.run
Submitted job id is 3998412.chadmin1
Submit job case.st_archive
Submitting job script qsub -q share -l walltime=0:20:00 -A P93300641 -W depend=afterok:3998412.chadmin1 -v ARGS_FOR_SCRIPT='--resubmit' case.st_archive
Submitted job id is 3998413.chadmin1
Submitted job case.run with id 3998412.chadmin1
Submitted job case.st_archive with id 3998413.chadmin1
```

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

You will probably want to check on the status of your jobs

Checking jobs:

Type '*qstat*'

```
clm_tutorial_cases/I1850CLM50_001> qstat
Job id            Name             User             Time Use S Queue
----------------  ---------------- ---------------  -------- - -----
3998413.chadmin1  I1850CLM50_001.  dll                     0 H shareex
3998412.chadmin1  I1850CLM50_001.  dll                     0 Q regular
clm_tutorial_cases/I1850CLM50_001>
```

Short-term archive

Simulation

Job ID

If you want to stop the simulation, you will need to kill your job.

Killing jobs:

a. Find your Job ID after typing *qstat*

b. Type '*qdel <Job ID>*'

Outline

Lecture/Intro

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

Practical

- **Download CLM code** (this is a one time setup step)
- **Basic workflow**
  1) Create a new case
  2) Invoke case_setup
  3) Build the executable
  4) Run and output data

- **Finding & Looking at model output**

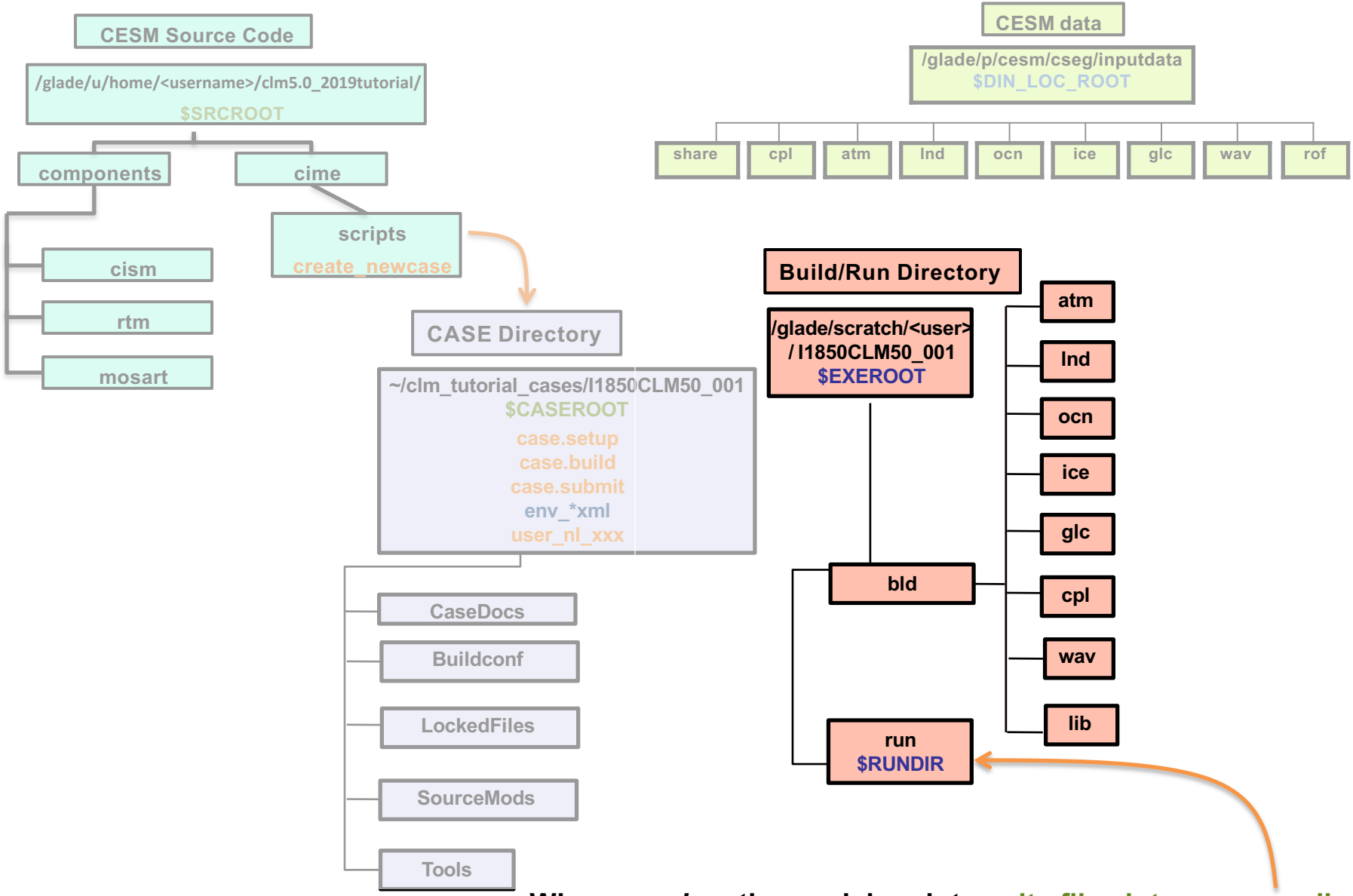
Other Useful Info

- **Getting help**

Where is the model output?

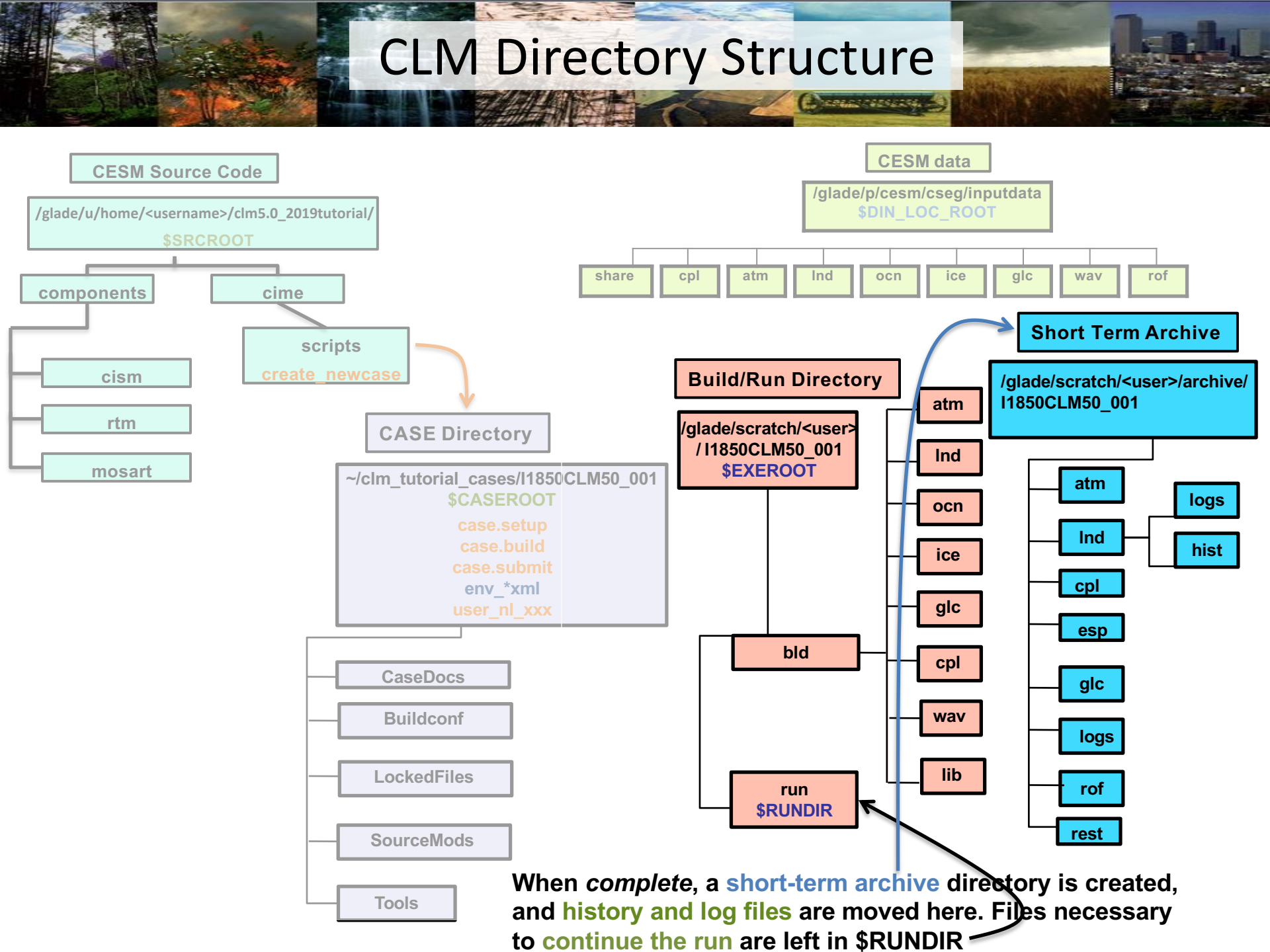
Your simulation will likely take some time to complete. The information provided next shows where the model output will be located while the model is running and once the simulation is complete. We also provide files from a simulation that is already complete so that you can do the next exercises before your simulation completes.

CLM Directory Structure



When *running*, the model scripts write files into your run directory.

CLM Directory Structure



Finding model output

When the simulation is ***complete***, a short-term archive directory is created, and history and log files are moved here.

Note: you will not see this directory until both your simulation and the short-term archive script have finished running

/glade/scratch/*<username>*/archive/I850CLM50_001/lnd/hist

Change this to your user name

This is your case name

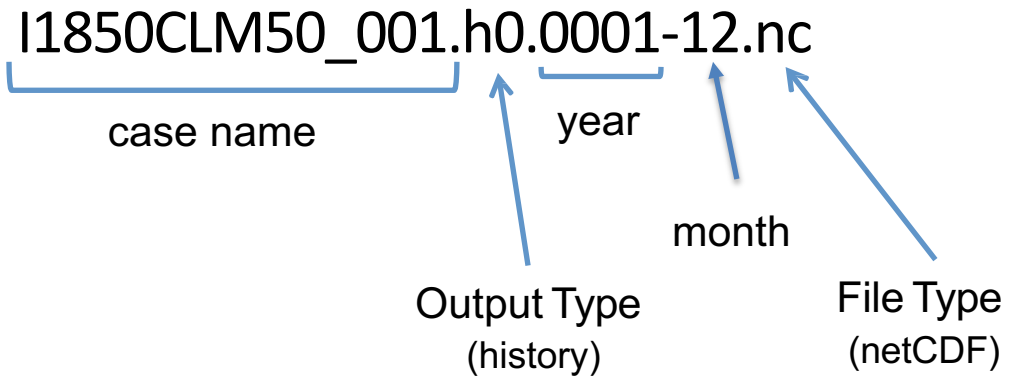
Note that files necessary to continue the run are left in the run directory: /glade/scratch/<username>/<casename>/run.

Finding model output

When the simulation is ***complete***, a short-term archive directory is created, and history and log files are moved here.

/glade/scratch/<*username*>/archive/I850CLM50_001/lnd/hist

Files (use "*ls*" to list them):



Note that files necessary to continue the run are left in the run directory: /glade/scratch/<username>/<casename>/run.

Finding model output

When the simulation is ***complete***, a short-term archive directory is created, and history and log files are moved here.

/glade/scratch/<*username*>/archive/I850CLM50_001/lnd/hist

Your simulation is likely still in the queue (check using `qstat`)

Check again before you leave today to see if your simulation completed and if the files were transferred here.

Next, let's explore data from a similar simulation that already ran

Looking at model output

There are a few command-line tools you can use to view netCDF data files. Two of the most useful:

ncdump This is a tool that generates a text representation of netCDF data. It is useful for providing information about the variables (names, types, and shapes), dimensions (names and sizes), attributes (names and values), and values of data for all or selected variables.

ncview This is a browser designed to view visualization of netCDF data, displaying a 2-dimensional color representation of data. It is useful for looking at data across various dimensions using a quick and easy push-button interface.

Let's make sure your login environment is set up to use these tools

---

To Do:

First, check the system modules that are loaded in your login environment:

```
module list
```

Do you see the modules "netcdf" and "ncview"? If not, do the following:

```
module load netcdf
module load ncview
```

Note: if you are having trouble loading these modules, it may be because you do not have a compiler loaded. Try: `module load Intel`

Looking at model output

To Do:

Navigate to this directory, where data from a completed simulation are stored:

```
cd /glade/p/cgd/tss/CTSM_tutorial2019/Practical1/I1850CLM50
```

Let's look at the information included in the file in a text format

```
ncdump -h I1850CLM50_001.clm2.h0.0001-01.nc |more
```

Notes:

1) *Use the "`-h`" option to look through the variable names, attributes and dimensions. If you do not use an option, ncdump will list this information and all the data values of all the variables, which is a lot of information!!*
2) *Use the "`|more`" so that you can scroll through the information from the start of the file.*

Use the spacebar to scroll through the file information. If you want to exit, scroll to the bottom or type "q"

Next, let's look at a map of the raw data

```
ncview I1850CLM50_001.clm2.h0.0001-*.nc &
```

Notes:

1) *Here, I use the wildcard, "`*`", option instead of a specific month to look at all the months in this year.*
2) *Use the "`&`" to send this program to the background. This way you can still use the command-line.*
3) *Ncview pops up a new window, so requires an x-forwarding option, such as XQuartz for Mac*

This will pop up an interface where you can select a variable to view. Play around with the different buttons & options to look at data in different ways. You can also double-click on a single point and it will generate a timeseries plot.

Looking at model output

While command-line viewing options like ncdump and ncview are useful, they are limited. Other options for looking at data from the model include:

- using the developed postprocessing tools (more on this tomorrow)
- Writing code to read in and analyze the netCDF files. Many languages will work: R, Python, NCL, IDL, Matlab, etc.

Next, we will use python code within Jupyter Lab to do some basic analysis of model output

Basic Analysis using Jupyter Lab

Jupyter Lab is an open-source web application that allows you to put live code, code output, plots, and narrative text into the same document.

One time setup: Download Tutorial Jupyter Labs

To Do:

Navigate to your home directory:

```
cd ~
```

Copy the ctsm_tutorial_jupyter directory into your home directory:

```
cp -r /glade/p/cgd/tss/CTSM_tutorial2019/ctsm_tutorial_jupyter .
```

Note: the `-r` option used here stands for "recursive", and will copy the entire directory

Next, let's use Jupyter Lab to look at CLM output

Basic Analysis using Jupyter Lab

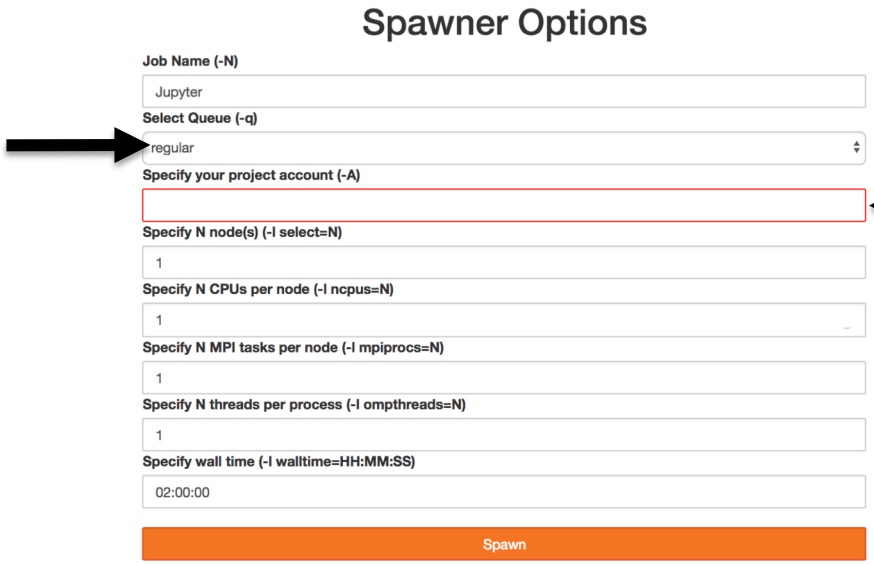
Using a web browser, navigate to this website:

http://jupyterhub.ucar.edu/ch

Log into the website using your cheyenne log in credentials: your username and yubikey (as the password).

This will navigate to a webpage with "Spawner Options". You'll need to enter the JupyterHub project number and then click "Spawn"

Enter Queue number `R4230874` here



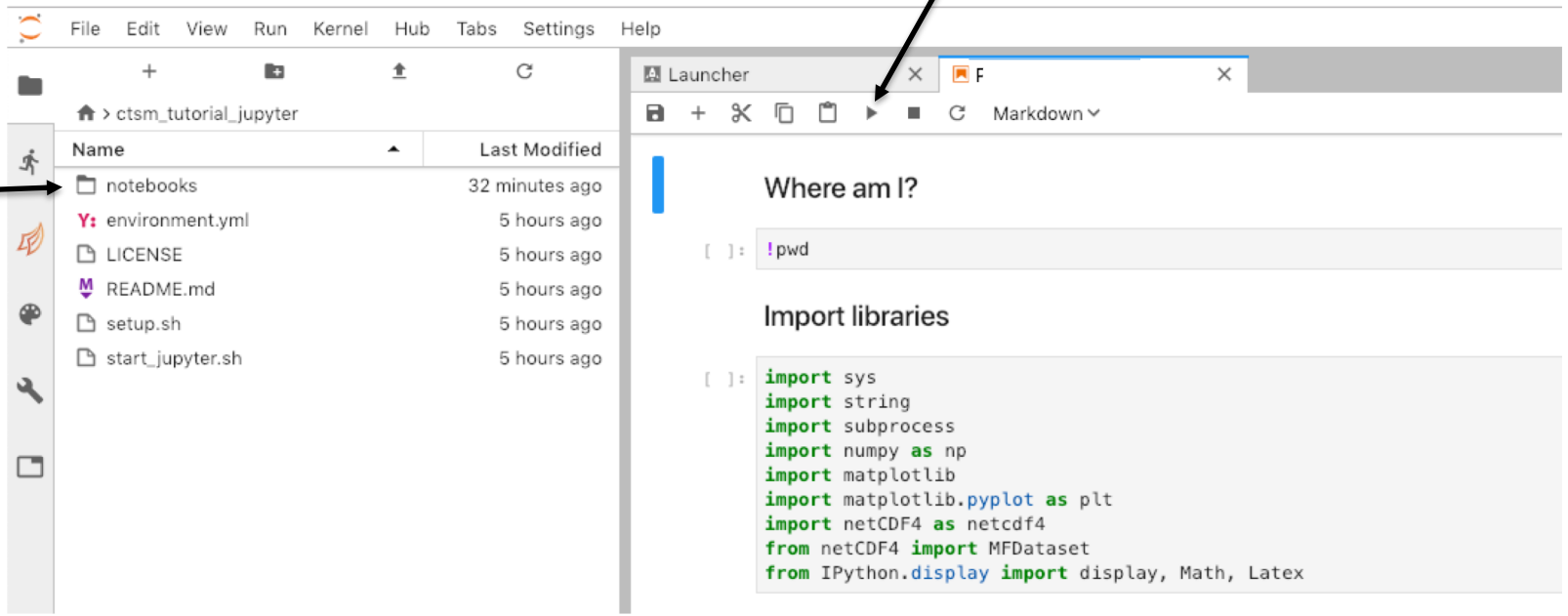
Enter Project number UCGD0004 here

Basic Analysis using Jupyter Lab

This will take you a new page that will allow you to explore the CLM output using code we developed in Jupyter Lab. On the left, navigate to the "notebooks" directory, and then click on "Practical1". This will bring up the lab, with sections of code and comments on the left. Run through the exercises here. You can use the ▶ on the navigation bar or use the keyboard commands "shift" + "return"

This button runs a block of code and advances to the next cell

Navigate to the "notebooks" folder to find Practical1.ipynb



Note: when you are finished with Jupyter, close the browser window and use "ctrl" + "c" in the terminal to exit and bring up a new command line

To Recap, today you learned skills to:

- Set up and run an out-of-the-box CLM5 simulation
- Use XML tools to customize a simulation
- Explore model output

Congratulations on learning these new skills! Tomorrow, you will use these again and learn some basic model configuration changes.

Outline

Lecture/Intro

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

Practical

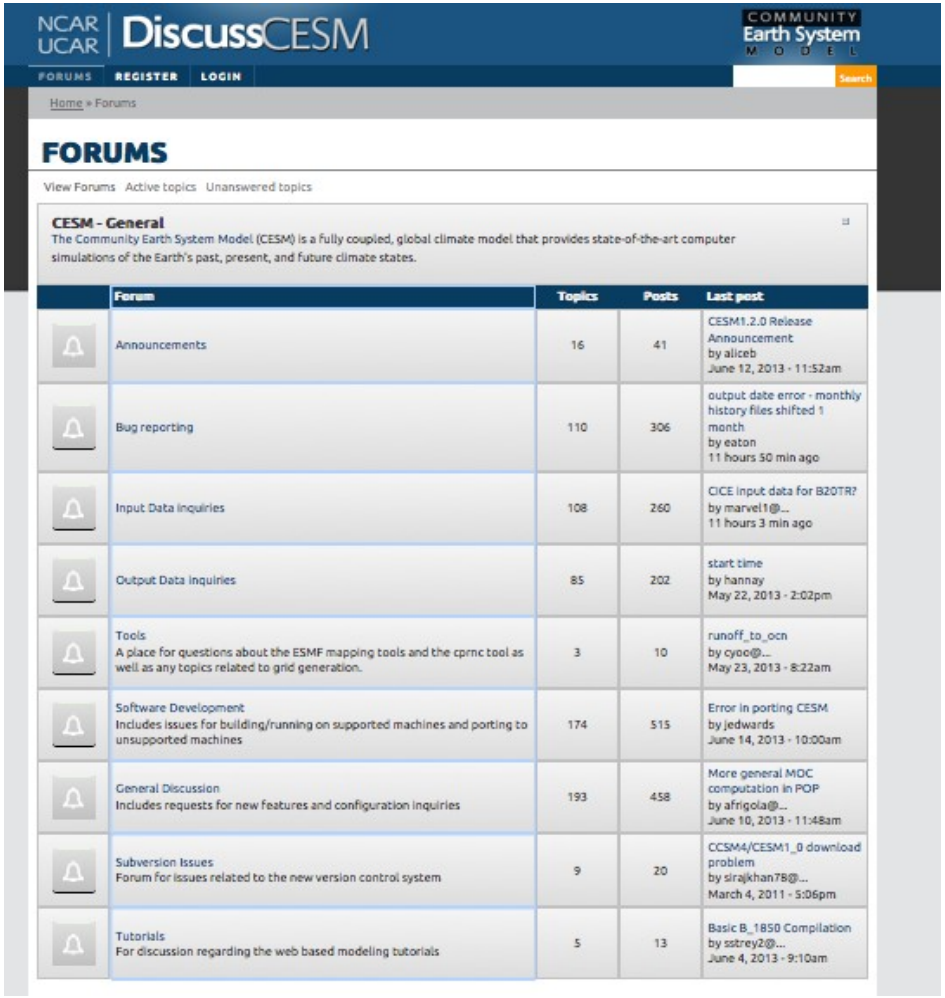
- **Download CLM code** (this is a one time setup step)
- **Basic workflow**
  1) Create a new case
  2) Invoke case_setup
  3) Build the executable
  4) Run and output data
- **Finding & Looking at model output**

Other Useful Info

- **Getting help**

Getting Help

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



- **Register** as a forums user by entering your information
- **Subscribe** to forums of interest. We recommend the announcements and known problems forums
- **Join** the CESM participants email list: http://mailman.cgd.ucar.edu/mailman/listinfo/ccsm-participants
- **Create** a github account and ”watch” CLM-related repositories