CESM Tutorial

NCAR Climate and Global Dynamics Laboratory

CESM 2.0
CESM1.2.x and previous (see earlier tutorials)

Alice Bertini

NCAR is sponsored by the National Science Foundation
Outline

• The CESM webpage

• Software & Hardware Requirements

• One-Time Setup

• Creating & Running a Case

• Getting More Help

• 7th Inning Stretch

• Review of Hands-on Exercises
CESM Models | CESM Supported Releases

You should use the most recent version of the model that is available unless you are trying to replicate previous results or create a branch run from a previous experiment. A complete list of CESM scientifically validated configurations is available for users needing to run the model in one of these configurations.

This table lists the most current supported CESM release versions.

<table>
<thead>
<tr>
<th>Supported CESM</th>
<th>Release Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CESM 1.2.x</td>
<td>Release Notes</td>
</tr>
<tr>
<td></td>
<td>Includes: What's New - Science,</td>
</tr>
<tr>
<td></td>
<td>What's New - Software, Answer-</td>
</tr>
<tr>
<td></td>
<td>Changing Features, Supported</td>
</tr>
<tr>
<td></td>
<td>Machines, and Known Problems</td>
</tr>
<tr>
<td>CESM 1.1.x</td>
<td>Notable Improvements</td>
</tr>
<tr>
<td>CESM 1.0.x</td>
<td>Notable Improvements</td>
</tr>
</tbody>
</table>

CESM Model Version Naming Conventions

CESM X.Y.Z - CESM model release versions include three numbers separated by a dot (.) where:

- X - corresponds to the major release number indicating significant science changes.
- Y - corresponds to the addition of new infrastructure and new science capabilities for targeted components.
- Z - corresponds to release bug fixes and machine updates.

Each release includes the complete collection of component model source code, documentation, and input data. For model output data, see the Experiments and Output Data section of this website.

Users should read the CESM Data Management & Distribution Plan which documents the procedures for the storage and distribution of data associated with the CESM project.

A note about scientifically validated configurations and which release version of the CESM to use for your experiments

Scientific validation of the CESM 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.

CESM 1.2.x is in the process of adding scientifically validated configurations and these will be posted on the web site as they become available.

Component sets and resolutions are backward compatible with all CESM releases. However, newer releases of the CESM allow for additional compsets, resolutions and machines.

The DiscussCESM Forums bulletin board can also provide specific recommendations from the CESM community regarding which release of the model to use for your specific requirements.
In Development - CESM2

About CESM2

TO DO Brief Description of CESM2
- What's New in CESM2
- CESM2 Supported Release Tags and 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.

- Experiment Diagnostics
- Experiment Output Datasets on the Earth Systems Grid
- Experiment Case Naming Conventions
- Experiment Output File Naming Conventions

Quick Start Documentation

- CESM2 Quick Start Guide
- TODO - move these into quick start guide User Workflows and Examples
- Register and Download
- Getting Help - DiscussCESM Forums

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).

Related Information

- Downloading the CESM Code
- CESM Data Management & Distribution Plan
- CESM Development Project Policies & Terms of Use
- CESM Support Policy
- DiscussCESM Forums Bulletin Board
CIME - Documentation

Common Infrastructure for Modeling the Earth (CIME) contains the support scripts (configure, build, run, test), data models, essential utility libraries, a "main" and other tools that are needed to build a single-executable coupled Earth System Model. CIME is available in a stand-alone package that can be compiled and tested without active prognostic components but is typically included in the source of a climate model. CIME does not contain any active components, any intra-component coupling capability (such as atmosphere physics-dynamics coupling).

- Common Infrastructure for Modeling the Earth (CIME) User's Guides
  Includes CIME, Driver-Coupler and Data Models Documentation

Active or Prognostic Components

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

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

All CESM2.0 Component Configurations

TODO - update just prior to release

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

- Component sets (compsets) defined by prognostic components
- Component Fortran Namelist settings
- Component XML variable definitions
- TODO - move these into Component Sets (compsets)
- Component Namelists and XML Variables Definitions

Model Grids and Machines

TODO - update just prior to release

- Grid Resolutions
- Supported Machines
Performance Data

- Performance and Load Balancing Data
- Running CESM2 on a Medium-sized Linux Cluster

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! Access to these external python based tools are being provided to the community via NCAR Github repositories.

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 inputdata repository has much more data in it than you need to run CESM ---- **DO NOT attempt to svn checkout the whole input data repository**. The **CIME User's Guide** explains how to obtain the subset of input data required for your needs.
Hardware/Software Requirements

- **Supported platforms**

  CESM2.0 currently runs “out of the box” today on the following machines
  - cheyenne – NCAR SGI
  - yellowstone – NCAR IBM
  - hobart – NCAR medium sized Linux cluster
  - edison / cori – NERSC Cray XC
  - pleiades – NASA SGI ICE cluster

  Always review the model version release notes and DiscussCESM Forums for up-to-date machine specific issues.

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

- **Running CESM2.0 on other platforms**

  Require porting + software
  - Subversion client (version 1.8 or greater)
  - python 2.7 and perl 5
  - Fortran and C compilers (recommend pgi, intel, or gnu compilers)
  - NetCDF library (recommend netcdf4.4 or later)
  - pnetcdf
  - MPI (MPI1 is adequate, Open MPI or MPICH seem to work on Linux clusters)
  - CMake
Basic Work Flow
(or how to set up and run an experiment)

• One-Time Setup Steps
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• Creating & Running a Case
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Please register as an individual CESM user even if your institution has a common installation of CESM.
Basic Work Flow
(or how to set up and run an experiment)

• **One-Time Setup Steps**
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• **Creating & Running a Case**
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
(B) Download the Source Code

• Code and input datasets are in Subversion repositories (*)
  https://svn-ccsm-release.cgd.ucar.edu/model_versions
  https://svn-ccsm-models.cgd.ucar.edu/cesm1/release_tags

• List the versions available on the CESM repository
  svn list https://svn-ccsm-models.cgd.ucar.edu/cesm1/release_tags

• Check out a working copy from the repository ("Download code")
  svn co https://svn-ccsm-models.cgd.ucar.edu/cesm1/release_tags/cesm1_2_2_1

• NOTE: For CESM2 (when it is released) the svn command will be:
  svn co https://svn-ccsm-models.cgd.ucar.edu/cesm2/release_tags/cesm2_0_0

(*) You can get subversion at http://subversion.apache.org/
Overview of Directories
(after initial model download)

The CESM2 source root contains 3 important directories: components, cime, and cime_config.

New in CESM2 - CIME (pronounced “SEAM”)
Separates the model infrastructure from the prognostic or “active” model components. This separation allows the option to “plug-and-play” different component models.

Github Public Repository: http://github.com/ESMCI/cime
Coupling Infrastructure for Modeling Earth (CIME)  
(new python-based CESM infrastructure)

- Infrastructure
  PUBLIC Open Source Github Repository

- Paradigm for DOE, NOAA, NSF Infrastructure Collaborations
  Driver-Coupler
  Data Models
  Scripts
  System/Unit testing
  Mapping Utilities

- Science code
  Restricted or Public Repositories

- CESM
- MPAS/WRF
- DOE/ACME
- ESPC and/or NOAA/NEMS

addresses needs of multiple efforts
Current CESM2 Coupling – data components permit flexible activation/deactivation of feedbacks

- River
  - DROF
  - MOSART
- Land Ice
  - CISM
- Ocean
  - DOCN
  - POP2
- Sea Ice
  - DICE
  - CICE5
- Land
  - DLND
  - CLM5
- ATM
  - DATM
  - CAM6
- MCT
- ESP
  - DESP
  - DART
- WAVE
  - DWAV
  - WW3

MCT – Model Coupling Toolkit
For software engineers: CIME uses XML files as the data store for configuration and variable settings and a set of python modules to parse those XML files and create an experiment case specific environment for setup, build, and batch submission.
Basic Work Flow
(or how to set up and run an experiment)

• One-Time Setup Steps
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• Creating & Running a Case
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Overview of Directories (+ inputdata directory)

Inputdata directory \$DIN_LOC_ROOT contains all input data required to run the model.
- on supported machines - populated inputdata already exists
- on non-supported machines - need to create inputdata directory root

- Ideally directory is shared by a group of users to save disc space
- To download input data: use the script check_input_data which is copied into every caseroot directory.
  - downloads only the data needed
  - puts the data in the proper subdirectories
  - Do NOT download input data manually (ie. by using svn co)
Basic Work Flow
(or how to set up and run an experiment)

• **One-Time Setup Steps**
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• **Creating & Running a Case**
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
(D) Porting

Porting details will be covered in Wednesday’s 1:00 p.m. lecture

CIME Documentation Part 2 – http://esmci.github.io/cime/

- On supported machines - no porting is necessary
- On new machines - porting needs to be done

From the CESM2 webpage:

NCAR's Experience Porting and Running CESM2 on a Medium-sized Linux Cluster

NCAR typically runs CESM on large super-computers with 4096 cores on yellowstone and 2160 cores on cheyenne. However, we also port, run and regularly tested CESM on a more moderately-sized Linux cluster.

NCAR's Climate and Global Dynamics (CGD) division maintains a medium-size Linux cluster called hobart to support research and development.

This page details our experiences on hobart that might help other institutions port and run CESM2 on their Linux clusters.

*NOTE* This is for information purposes only. Please use the DiscussCESM forums to post your questions regarding porting and running on your particular Linux cluster.

**Linux Cluster Hardware Specifications**

**Single login node with the following specifications:**

<table>
<thead>
<tr>
<th>Hostname</th>
<th>hobart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>CentOS Linux release 7.2.1511 (Core) x86_64</td>
</tr>
<tr>
<td>Kernel</td>
<td>3.10.0-327.el7.x86_64</td>
</tr>
<tr>
<td>Processor(s)</td>
<td>16 X Intel(R) Xeon(R) CPU WS580 @ 3.20GHz</td>
</tr>
<tr>
<td>CPU MHz</td>
<td>3192.072</td>
</tr>
<tr>
<td>Total Memory</td>
<td>74.05 GB</td>
</tr>
<tr>
<td>Total Swap</td>
<td>1.04 GB</td>
</tr>
</tbody>
</table>

**32 compute nodes with the following specifications for each node:**

| Operating System | CentOS Linux release 7.2.1511 (Core) x86_64 |
| Kernel | 3.10.0-327.el7.x86_64 |
| Processor(s) | 48 X Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz |
| CPU MHz | 23000.000 |
| Total Memory | 98.59 GB |
| Total Swap | 1.04 GB |

Available shared disk space for run and build directories: 5.0 T
Basic Work Flow
(or how to set up and run an experiment)

• One-Time Setup Steps
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• Creating & Running a Case
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Work Flow: Super Quick Start

CESM2 can be run with a set of 4 commands

Set of commands to build and run the model on a supported machine

1. Go into the directory “cases” in your home directory.
   ```bash
   mkdir ~/cases
   cd /glade/p/cesm/tutorial/cesm2_0_alpha07c/cime/scripts
   ./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850
   ```

2. Go into the case you just created.
   ```bash
   cd ~/cases/b.day1.0
   ```

3. Invoke case.setup.
   ```bash
   ./case.setup
   ```

4. Build the executable.
   ```bash
   ./case.build
   ```

5. Submit your run to the batch queue.
   ```bash
   ./case.submit
   ```

It is that easy!
Overview of Directories (+ before create_newcase)

CESM Download

~/cesm2_0_0 $SRCROOT

components

cime_config

cime/scripts
create_newcase
query_config

INPUTDATA Directory

/glade/p/cesm/cseg/inputdata
$DIN_LOC_ROOT

share cpl atm Ind ocn ice glc wav rof

This is the **script you need** to create a new case

create_newcase --help

Helper script `query_config` allows you to view the available XML configuration settings.

query_config --help
Set of commands to build and run the model on a supported machine

# one time step
mkdir ~/cases

# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm2_0_alpha07c/cime/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850

# go into the case you just created in the last step
cd ~/cases/b.day1.0/

# (2) invoke case.setup
./case.setup

# (3) build the executable
./case.build

# (4) submit your run to the batch queue
./case.submit
(1) 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?
Which resolution?
Which model configuration?
Which set of components?

NOTE: CESM2 no longer requires the `--mach` argument when running on supported machines.
(1) create_newcase arguments

create_newcase requires 3 arguments

create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850

**NOTE:** for all user scripts, you can run the script name followed by the **--h** or **--help** argument to see help documentation and a list of all command line arguments.

**NOTE:** Double dashes “--” are now required with command line arguments!
(1) create_newcase arguments

create_newcase requires 3 arguments

create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850

What is the casename?

case specifies the name and location of the case being created
~/cases/b.day1.0

NOTE: experiment case naming conventions for CESM are described on the CESM2 webpage at URL:
http://www.cesm.ucar.edu/models/cesm2.0/cesm/casename_conventions_cesm.html
(1) create_newcase arguments

create_newcase requires 3 arguments

```
create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850
```

Which resolution?

res specifies the model resolution (or grid)

New 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
**Grid Resolution Definitions**

Model Version: CESM2.0  
HTML created on: 2017-05-12

This page contains the complete list of grid resolution short names and descriptions. They are grouped by alias names designed to aid reading. Clicking on the blue text will display additional descriptive information. Click on the "Show Details" button and then ctrl+F key to search for specific strings in this file.

<table>
<thead>
<tr>
<th>Grid Naming Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Default Component Grids</strong></td>
</tr>
<tr>
<td>Grid Alias: 1D_1D (only for compsets that are DATM.+DROF)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_brazil (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_camdenNJ (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_mexicocityMEX (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_numaLA (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_smallvilleIA (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_tropicAtl (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_urbanc_alpha (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 1x1_vancouverCAN (only for compsets that are DATM.+CLM)</td>
</tr>
<tr>
<td>Grid Alias: 5x5_amazon (only for compsets that are DATM.+CLM)</td>
</tr>
</tbody>
</table>

Non-default grids are: atm:5x5_amazon lnd:5x5_amazon rof:NULL

5x5 Amazon regional case -- only valid for DATM/CLM compset with domain file(s):
$CGN_ROOT/share/domains/domain.clm/domain.lnd.5x5pt-amazon_navy.090715.ne (only for grid match: atm|lnd)

| Grid Alias: CLM_USRDAT (only for compsets that are DATM.+CLM) |
| Grid Alias: T31_g37 |
| Grid Alias: T31_g37_gl10 (only for compsets that are _CISM) |
(1) create_newcase arguments

create_newcase requires 3 arguments

create_newcase --case ~/cases/b.day1.0 --res T31_g37 --compset B1850

Which component set?

compset specifies the “component set”

Component set specifies component models, forcing scenarios and physics options for those models

New compset naming convention
Each model compset can be specified by its alias or long name.
Example of equivalent alias, short name and long name:
- alias: B1850
- long name = 1850_CAM60_CLM50%BGC_CICE_POP2%ECO_MOSART_CISM2%NOEVOLVE_WW3_BGC%BDRD

↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑
time atm Ind ice ocn river Ind-ice wave BGC scenario
## CESM2 Supported Component Set (compset) Definitions

[http://www.cesm.ucar.edu/models/cesm2.0/cesm/grids.html](http://www.cesm.ucar.edu/models/cesm2.0/cesm/grids.html) query_config – compsets [{all, allactive, drv, cam, cism, clm, cice, pop, mpas-o}]

### Component Set Definitions (compsets)

Model Version: CESM2.0
HTML created on: 2017-06-04

This page contains the complete list of component sets aliases and long names. They are grouped by model components designed to aid browsing.

Clicking on the name of a component will display additional descriptive information. Click on the “Show Details” button and then ctrl+F key to search for specific strings in this file.

#### Compset Naming Convention

**Component: allactive**

<table>
<thead>
<tr>
<th>Alias</th>
<th>Long Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1850</td>
<td>1850_CAM50_CLM50%BGG_CIGE_CICE_POP2_ECO_MOSART_CISM2_NO_EVOLVE_WW3_BGC_BRD</td>
</tr>
<tr>
<td>B1850CL45BGCBPRP</td>
<td>1850_CAM40_CLM45%BGG_CIGE_CICE_POP2_ECO_MOSART_SGLC_SAV_WBG_BGC_BPRP</td>
</tr>
<tr>
<td>B1850CL45BGCRPRP</td>
<td>1850_CAM40_CLM45%BGG_CIGE_CICE_POP2_ECO_RTM_SGLC_SAV_WBG_BPRP</td>
</tr>
<tr>
<td>B1850CL45BGCG</td>
<td>1850_CAM50_CLM45%BGG_CIGE_CICE_POP2_MOSART_SGLC_SAV</td>
</tr>
<tr>
<td>B1850CL45BGCR</td>
<td>1850_CAM50_CLM45%BGG_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1850CL45BGCRPRP</td>
<td>1850_CAM50_CLM45%BGG_CIGE_CICE_POP2_ECO_RTM_SGLC_SAV_WBG_BPRP</td>
</tr>
<tr>
<td>B1850CL45BGCRS</td>
<td>1850_CAM50_CLM45%BGG_CIGE_CICE_POP2_RTM_CISM_SAV</td>
</tr>
<tr>
<td>B1850CL45BGCRG</td>
<td>1850_CAM50_CLM45%BGG_CIGE_CICE_POP2_RTM_CISM1_SAV</td>
</tr>
<tr>
<td>B1850W1s</td>
<td>1850_CAM50_CLM50%BGG_CIGE_CICE_POP2_ECO_MOSART_CISM2_NO_EVOLVE_SAV_WBG_BRD</td>
</tr>
<tr>
<td>B1850W1sGC</td>
<td>2000_CAM50_CLM50%BGG_CIGE_CICE_POP2_RTM_CISM_SAV</td>
</tr>
<tr>
<td>B1850W1sGCG</td>
<td>2000_CAM50_CLM50%BGG_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1850W1sGCGR</td>
<td>2000_CAM50_CLM50%BGG_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1850W1sGCGRG</td>
<td>2000_CAM50_CLM50%BGG_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>BHIST</td>
<td>HIST_CAM50_CLM50%BGG_CIGE_CICE_POP2_ECO_CISM2_NO_EVOLVE_SAV_BGC_BRD</td>
</tr>
<tr>
<td>BHHISTW1s</td>
<td>HIST_CAM50_CLM50%BGG_CIGE_CICE_POP2_ECO_CISM2_NO_EVOLVE_SAV_BGC_BRD</td>
</tr>
<tr>
<td>B1P85CL45BGGC</td>
<td>RCP8_CAM50_CLM45%BGG_CIGE_CICE_POP2_MOSART_SGLC_SAV</td>
</tr>
<tr>
<td>B1P85CL45BGCR</td>
<td>RCP8_CAM50_CLM45%BGG_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1P85CL45BGBC</td>
<td>RCP8_CAM50_CLM45%BGG_CIGE_CICE_POP2_ECO_MOSART_SGLC_SAV</td>
</tr>
<tr>
<td>B1P85CL45BGBB</td>
<td>RCP8_CAM50_CLM45%BGG_CIGE_CICE_POP2_ECO_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>BW1850</td>
<td>1850_CAM60_WCTS_CLM50%BGG_CIGE_CICE_POP2_ECO_MOSART_CISM2_NO_EVOLVE_SAV_BGC_BRD</td>
</tr>
<tr>
<td>B1P85C5</td>
<td>1850_CAM50_W50all_CLM40%SP-WISO_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1P85C5CN</td>
<td>1850_CAM50_W50all_CLM40%CN-WISO_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1P85C5CN1</td>
<td>1850_CAM50_W50all_CLM40%CN-WISO_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>B1P85C5CN5</td>
<td>1850_CAM50_W50all_CLM40%CN-WISO_CIGE_CICE_POP2_RTM_SGLC_SAV</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
<tr>
<td>E1P85CL45TEST</td>
<td>1850_CAM45%SP_CIGE_DOCNOSM_MOSART_SGLC_SAV_TEST</td>
</tr>
</tbody>
</table>

### Compset Mode Descriptions

The modes define options for the different component models using a Python regular expression syntax for matching when creating a new case.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%ADIAB</td>
<td>CAM adiabatic physics:</td>
</tr>
<tr>
<td>%CCTS[ ]</td>
<td>CAM-Chem troposphere / stratosphere chemistry with simplified volatility basis set SOA scheme and modal aerosols:</td>
</tr>
<tr>
<td>%CLB[ ]</td>
<td>CAM CLUBB</td>
</tr>
<tr>
<td>%DABP04</td>
<td>CAM dry adiabatic baroclinic instability (Polven 2004).</td>
</tr>
<tr>
<td>%DCTBM</td>
<td>CAM dynamical core test with baroclinic wave IC and terminator chemistry:</td>
</tr>
<tr>
<td>%HS94</td>
<td>CAM Held-Suarez forcing:</td>
</tr>
<tr>
<td>%KESSLER</td>
<td>CAM dynamical core test with baroclinic wave IC and Kessler physics:</td>
</tr>
<tr>
<td>%PM[ ]</td>
<td>CAM prescribed modal aerosols:</td>
</tr>
<tr>
<td>%PORT</td>
<td>CAM Parallel Offline Test:</td>
</tr>
<tr>
<td>%RGO2[ ]</td>
<td>CAM CO2 ramp:</td>
</tr>
<tr>
<td>%SPCAMCLBM</td>
<td>CAM super-parameterized CAM double moment m2005 SAM microphysics using CLU BB</td>
</tr>
<tr>
<td>%SPCAMCLBS</td>
<td>CAM super-parameterized CAM one moment SAM microphysics using CLU BB</td>
</tr>
<tr>
<td>%SPCAMM</td>
<td>CAM super-parameterized CAM double moment m2005 SAM microphysics</td>
</tr>
<tr>
<td>%SPCAMSM</td>
<td>CAM super-parameterized CAM one moment SAM microphysics</td>
</tr>
<tr>
<td>%TMOZ[ ]</td>
<td>CAM tropospheric chemistry with bulk aerosols:</td>
</tr>
<tr>
<td>%WCCM[ ]</td>
<td>CAM WACC with middle atmosphere chemistry:</td>
</tr>
<tr>
<td>%WCGS[ ]</td>
<td>CAM WACC specified chemistry:</td>
</tr>
<tr>
<td>%WCTS[ ]</td>
<td>CAM WACC with tropospheric, stratospheric, mesospheric, and lower thermospheric chemistry:</td>
</tr>
<tr>
<td>%WXIEP %1</td>
<td>CAM WACC-X enhanced:</td>
</tr>
</tbody>
</table>
Result of running create_newcase

```
./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850
```

```
Compset longname is 1850_CAM60_CLM50%BG-CROP_CICE_POP2%ECO_MOSART_CISM2%NOEVOLVE_WW3_BGC%BDRD
Compset specification file is /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/cime/..../cime_config/config_compsets.xml
Compset forcing is 1850
Com forcing is Biogeochemistry intercomponent
ATM component is CAM cam6 physics:
LND component is clm5.0 bgc with prognostic crop:
ICE component is prognostic cice:
OCN component is POP2 default:POP2/Ecosystem:
ROF component is MOSART:
GLC component is cism2:cism ice evolution turned off (this is the standard configuration unless you're explicitly interested in ice evolution):
WAV component is WW3:
ESP component is
Pes specification file is /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/cime/..../cime_config/config_pes.xml
Machine is cheyenne
Pes setting: grid match is a%1.9x2.5+l%1.9x2.5+oi%gx1
Pes setting: grid is a%1.9x2.5+l%1.9x2.5_oigx1v7_r%r05_gland4_w%ww3a_m%gxlv7
Pes setting: compset is 1850_CAM60_CLM50%BG-CROP_CICE_POP2%ECO_MOSART_CISM2%NOEVOLVE_WW3_BGC%BDRD
Pes setting: tasks is {NTASKS_ATM: -4, NTASKS_ICE: -2, NTASKS_CPL: -4, NTASKS_LND: -2, NTASKS_WAV: -1, NTASKS_S_ROF: -2, NTASKS_OCN: -2, NTASKS_GL: -1}
Pes other settings: {}  
Compset is: 1850_CAM60_CLM50%BG-CROP_CICE_POP2%ECO_MOSART_CISM2%NOEVOLVE_WW3_BGC%BDRD
Grid is: a%1.9x2.5_l%1.9x2.5_oigx1v7_r%r05_gland4_w%ww3a_m%gxlv7
Components in compset are: ['cam', 'clm', 'cice', 'pop', 'mosart', 'cism', 'ww3', 'sesp', 'cpl', 'dart']
This compset and grid combination is not scientifically supported, however it is used in 7 tests.

Using project from env PROJECT: P93300606
cesm model version found: cesm2_0_alpha07b
Creating Case directory /glade/u/home/aliceb/cases/b.day1.0
```

Success!
Overview of Directories (after create_newcase)

**CESM Download**
- `~/cesm2_0_0`
- `$SRCROOT`
- `components`
  - aquap
  - cam
  - cice
  - cism
  - clm
  - mosart
  - pop
  - rtm
  - ww3

**CASE Directory**
- `$CASEROOT`
- `case.setup`
- `env_*xml`
- `xmlchange`
- `~/cases/b.day1.0`

**CASE Directory contains:**
- `case.setup`: script used in the next step
- `files with xml variables used by CESM scripts`
- `script to edit env_*xml files`
- `subdirectory for case specific code modifications`

**INPUTDATA Directory**
- `/glade/p/cesm/cseg/inputdata`
- `$DIN_LOC_ROOT`
- `share`
- `cpl`
- `atm`
- `ind`
- `ocn`
- `ice`
- `glc`
- `wav`
- `rof`

**Build/Run Directory**
- `/glade/scratch/use rx/b.day1.0`
- `$EXEROOT`
- `bld`
- `run $RUNDIR`

**create_newcase**
- creates case directory that contains:
- `files with xml variables used by CESM scripts`
- `script to edit env_*xml files`
- `subdirectory for case specific code modifications`

**Create Newcase**
- creates case directory that contains:
- `components`
- `scripts`
- `create_newcase`
- `cime_config`
- `components`
- `cime/scripts`
- `create_newcase`

**Build/Run Directory**
- `/glade/scratch/use rx/b.day1.0`
- `$EXEROOT`
- `bld`
- `run $RUNDIR`

**Create Newcase**
- creates case directory that contains:
- `components`
- `scripts`
- `create_newcase`
- `cime_config`
- `components`
- `cime/scripts`
- `create_newcase`
About env_*_.xml files

env_*_.xml contains variables used by scripts -- some can be changed by the user

- 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 script case.submit
- env_build.xml: specifies build information used by script case.build
- env_case.xml: set by create_newcase and cannot be modified
- env_mach_pes.xml: specifies PE layout of components used by script case.run
- env_mach_specific.xml: specifies machine specific information used by script case.build
- env_run.xml: sets run time information (such as length of run, frequency of restarts, …)
  User interacts with this file most frequently

- To query a variable in an xml file use script xmlquery
- To modify a variable in an xml file use script xmlchange
  ./xmlchange STOP_N=20

NOTE: You can edit the XML files manually but it is recommended that you use the xmlchange script to ensure that the XML schema is preserved!
xmlchange

usage:
xmlchange --help
OR
xmlchange --test

EXAMPLES:
# xmlchange REST_OPT=ndays,REST_N=4
> xmlchange

This utility allows the user to change an env.*xml file via a command line interface. The command is echoed to the CaseStatus file, unless --noecho is given. The purpose of this echoing is to provide a "paper trail" of changes made by the user, so calls to xmlchange by the cime scripts that are part of the normal case setup/build process should generally use --noecho.

positional arguments:
listofsettings Comma separated list of settings in the form: var1=value,var2=value,... (default: )

optional arguments:
-h, --help show this help message and exit
-d, --debug Print debug information (very verbose) to file /glade/u/home/aliceb/cases/b.day1.0/xmlchange.log (default: False)
-v, --verbose Add additional context (time and file) to log messages (default: False)
-s, --silent Print only warnings and error messages (default: False)
--caseroot CASEROOT Case directory to change (default: /glade/u/home/aliceb/cases/b.day1.0)
--loglevel LOGLEVEL ignored, backward compatibility only (default: None)
-file FILE, --file FILE xml file to edit (default: None)
-id ID, --id ID the xml entry id (default: None)
-val VAL, --val VAL the value to set (default: None)
-delimiter DELIMITER, --delimiter DELIMITER set delimiter string, default is , (default: ,)
--dryrun DRYRUN, --dryrun DRYRUN parse settings and print key value pairs only (default: False)
--noecho, --noecho do not update CaseStatus with this change (default: False)
--append, --append append to the existing value (default: False)
--subgroup SUBGROUP, --subgroup SUBGROUP apply to this subgroup only (default: None)
-f, --force ignore typing checks and store value (default: False)
Basic Work Flow
(or how to set up and run an experiment)

• One-Time Setup Steps
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• Creating & Running a Case
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Work Flow: Super Quick Start

# go into scripts directory into the source code download
```bash
cd /glade/p/cesm/tutorial/cesm2_0_alpha07c/cime/scripts
```

# (1) create a new case in the directory “cases” in your home directory
```bash
./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850
```

# go into the case you just created in the last step
```bash
cd ~/cases/b.day1.0/
```

# (2) invoke case.setup
```bash
./case.setup
```

# (3) build the executable
```bash
./case.build
```

# (4) submit your run to the batch queue
```bash
./case.submit
```
About case.setup

```
./case.setup --help

Usage:
OR
case.setup --help
OR
case.setup --test

Examples:
  # Setup case
  > case.setup

case.setup - create the $caseroot/case.run script and user_nl_xxx component
  namelist mod files

Positional arguments:
caseroot Case directory to setup (default:
  /glade/u/home/aliceb/cases/b.day1.0)

Optional arguments:
  -h, --help show this help message and exit
  -d, --debug Print debug information (very verbose) to file
  /glade/u/home/aliceb/cases/b.day1.0/case.setup.log
  (default: False)
  -v, --verbose Add additional context (time and file) to log messages
  (default: False)
  -s, --silent Print only warnings and error messages (default: False)
  -c, --clean Removes the batch run script for target machine. If the
  testmode argument is present then keep the test script if
  it is present - otherwise remove it. The user_nl_xxx and
  Macros files are never removed by case.setup - you must
  remove them manually (default: False)
  -t, --test-mode Keeps the test script when the --clean argument is used
  (default: False)
  -r, --reset Does a clean followed by setup (default: False)
```

**NOTE:** changing any of the XML settings in env_mach_pes.xml requires
  ./case.setup --reset
Calling case.setup

[[aliceb@cheyenne6:b.day1.0]>./case.setup
/glade/u/home/aliceb/cases/b.day1.0/env_mach_specific.xml already exists, delete to replace
Creating batch script case.run
Writing case.run script from input template /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/cime/config/cesm/machines/template.case.run
Writing case.st_archive script from input template /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/cime/config/cesm/machines/template.st_archive
Locking file env_mach_pes.xml
Creating user_nl_xxx files for components and cpl
If an old case build already exists, might want to run 'case.build --clean' before building

[aliceb@cheyenne6:b.day1.0]>]

- Create $RUNDIR and $EXEROOT directories
- Create user_nl_xxx files
- Create scripts case.run, case.st_archive
- Create Macros.make file
- Create hidden files .env_mach_specific.* which can help with debugging
- Create CaseDocs directory - **NOTE:** these files should not be edited!
Overview of Directories (after case.setup)

CASE Directory

- ~cases/b.day1.0
- $CASEROOT
- case.setup
- case.build
- case.submit
- user_nl_xxx

SourceMods

Tools

Buildconf

CaseDocs

LockedFiles

Inputdata Directory

/glade/p/cesm/cseg/inputdata

$DIN_LOC_ROOT

share cpl atm Ind ocn ice glc wav rof

Inputdata Directory

Build/Run Directory

/glade/scratch/use

rx/b.day1.0

$EXEROOT

atm Ind ocn ice glc cpl rof wav

Build/Run directory

Buildconf

CaseDocs

LockedFiles

SourceMods

Tools

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

case.setup creates:

- case scripts (to build, run and archive)
- namelist modification files user_nl_xxx
  this is where you modify your namelists

CESM Download

~cesm2_0_0

$SRCROOT

components

cime_config

cime/scripts create_newcase

aquap
cam
cice
cism
clm
omart
pop
rtm
ww3

Build/Run directory

bld

run

$RUNDIR
Basic Work Flow
(or how to set up and run an experiment)

• One-Time Setup Steps
  (A) Registration
  (B) Download the CESM code
  (C) Create an Input Data Root Directory
  (D) Porting

• Creating & Running a Case
  (1) Create a New Case
  (2) Invoke case.setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Work Flow: Super Quick Start

Set of commands to build and run the model on a supported machine: ”cheyenne”

```bash
# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm2_0_alpha07c/cime/scripts

# (1) create a new case in the directory “cases” in your home directory
  ./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850CN

# go into the case you just created in the last step
  cd ~/cases/b.day1.0/

# (2) invoke case.setup
  ./case.setup

# (3) build the executable
  ./case.build

# (4) submit your run to the batch queue
  ./case.submit
```
Build the Model

- Modifications before build
  - Change env_build.xml values before running case.build
  - Introduce any modified source code in SourceMods/ before building

- To completely rebuild, run case.build –clean-all first

- The case.build script
  - Checks for missing input data
  - Builds the individual component libraries and model executable

- If any inputdata is missing,
  - Build aborts, but provides a list of missing files
  - Run ./check_input_data --download to acquire missing data
  - This will use svn to put required data in the inputdata directory
  - Then re-run build script
Running the case.build Script

- Checks for missing input data
- Aborts if any input data is missing

```
[alice@cheyenne6.b.day1.0]$ ./case.build
Building case in directory /glade/u/home/aliceb/cases/b.day1.0
sharedlib_only is False
model_only is False
Generating component namelists as part of build
Creating component namelists
  Calling /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/components/com/cime_config/buildml
...calling cam buildcpp to set build time options
  Calling /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/components/clm/cime_config/buildml
  Calling /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/components/cice/cime_config/buildml
...calling cice buildcpp to set build time options
  Calling /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/components/pop/cime_config/buildml
...calling pop buildcpp to set build time options
  Calling /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/components/mosart/cime_config/buildml
Running /glade/p/work/aliceb/sandboxes/dev/cesm2_0_alpha07/components/clm/cime_config/buildml

Finishing creating component namelists
Building gptl with output to file /glade/scratch/aliceb/b.day1.0/bld/gptl.bldlog.170806-153359
  Component gptl build complete with 9 warnings
Building mct with output to file /glade/scratch/aliceb/b.day1.0/bld/mct.bldlog.170806-153359
Building pio with output to file /glade/scratch/aliceb/b.day1.0/bld/pio.bldlog.170806-153359
Building csm_share with output to file /glade/scratch/aliceb/b.day1.0/bld/csm_share.bldlog.170806-153359
  Component csm_share build complete with 20 warnings
    - Building clm4.5/clm5.0 Library
Building lnd with output to /glade/scratch/aliceb/b.day1.0/bld/lnd.bldlog.170806-153359
  Component lnd build complete with 6 warnings
clm built in 175.854330 seconds
Building atm with output to /glade/scratch/aliceb/b.day1.0/bld/atm.bldlog.170806-153359
Building ice with output to /glade/scratch/aliceb/b.day1.0/bld/ice.bldlog.170806-153359
Building ocn with output to /glade/scratch/aliceb/b.day1.0/bld/ocn.bldlog.170806-153359
Building rof with output to /glade/scratch/aliceb/b.day1.0/bld/rof.bldlog.170806-153359
Building gld with output to /glade/scratch/aliceb/b.day1.0/bld/glc.bldlog.170806-153359
Building wav with output to /glade/scratch/aliceb/b.day1.0/bld/wav.bldlog.170806-153359
Building esp with output to /glade/scratch/aliceb/b.day1.0/bld/esp.bldlog.170806-153359
sres built in 1.398876 seconds
ww built in 154.859285 seconds
mosart built in 154.859462 seconds
  Component ice build complete with 1 warnings
  Component cice built in 427.852179 seconds
Component atm build complete with 14 warnings
Component gld build complete with 3 warnings
  Component ocn build complete with 4 warnings
pop built in 504.855526 seconds
clsm built in 504.855353 seconds
  Component atm build complete with 14 warnings
  Component gld build complete with 3 warnings
Component atm build complete with 14 warnings
Component ocn build complete with 4 warnings
pop built in 504.855526 seconds
clsm built in 504.855353 seconds
  cam built in 504.859419 seconds
Building cesm with output to /glade/scratch/aliceb/b.day1.0/bld/cesm.bldlog.170806-153359
Time spent not building: 15.649067 sec
Time spent building: 862.840315 sec
[alice@cheyenne6.b.day1.0]$]
```
Overview of Directories (after build)

**CASE Directory**
- b.day1.0
- $CASEROOT
cesm_setup
case.build
case.submit
- user_nl_xxx
- SourceMods

**Build/Run Directory**
- /glade/scratch/use rx/b.day1.0
  - $EXEROOT
- run
  - $RUNDIR

**INPUTDATA Directory**
- /glade/p/cesm/cseg/inputdata
  - $DIN_LOC_ROOT
  - share
  - cpl
  - atm
  - Ind
  - ocn
  - ice
  - glc
  - wav
  - rof

---

**CESM Download**
- ~/cesm2_0_0
  - **$SRCROOT**
  - CIME Download
    - components
      - aquap
      - cam
      - cice
      - cism
      - clm
      - mosart
      - pop
      - rtm
      - wW3
- cime_config
- cime/scripts
  - create_newcase

---

**The build script**
1. checks input data
2. Creates model executable and Fortran namelists

**If any inputdata is missing,**
- Build aborts and provides a list of missing files
- Run `./check_input_data --download` to get missing data
- Then re-run build script
Basic Work Flow
(or how to set up and run an experiment)

- **One-Time Setup Steps**
  - (A) Registration and Download
  - (B) Create an Input Data Root Directory
  - (C) Porting

- **Creating & Running a Case**
  - (1) Create a New Case
  - (2) Invoke cesm_setup
  - (3) Build the Executable
  - (4) Run the Model and Output Data Flow
Work Flow: Super Quick Start

Set of commands to build and run the model on a supported machine: ”cheyenne”

```bash
# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm2_0_alpha07c/cime/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850CN

# go into the case you just created in the last step
cd ~/cases/b.day1.0/

# (2) invoke case.setup
./case.setup

# (3) build the executable
./case.build

# (4) submit your run to the batch queue
./case.submit
```
(4) Running the Model

Check archive and Run options

Check if namelists need to be rebuilt

Load the initial input data

Submit case.st_archive dependent on the successful completion of case.run

Submit case.run

Batch job status

qstat –u aliceb
When running, the model scripts **write files into your run directory.**

After completion the **case.st_archive** will move files into the appropriate directories (next slide).
Overview of Directories (when the job completes) (archiving data)

(1) Move **timing** and **log** files into case directory
(2) Leave in $RUNDIR what is needed to continue the run
(3) Move history and log files to **short-term archive**
Expert feature: create_clone

• The “create_clone” tool copies an existing case to make a new copy.
• Things that are copied:
  • Most (not all) env_*.xml settings.
  • user_nl_xxx files
  • Macros
  • SourceMods
  • Batch system files
  • README.case
• Not copied:
  • Logs
  • Timing files

• Invocation (from cime/scripts directory):
  • ./create_clone --clone ~/cases/b.day1.0 --case ~/cases/b.day1.2
Best practices for copying cases

• Using “cp –R” does not work!
• When using create_clone, make sure that your changes will be minor:
  • Same version of the code!
  • Same grid
  • Same compset
  • Namelist/SourceMods changes not too complex.

• Document changes in your case directory so that they are easy to track: README.case is a great place.

• If your changes are more complex, if you use multiple code versions, or if you have to create a great many cases at once, consider writing your own script to set up your cases.
More Information/Getting Help

Model User Guides: http://www.cesm.ucar.edu/models/cesm2.0

Active or Prognostic Components

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

- Atmosphere
- Land
- Land Ice
- Ocean
- Sea Ice
- River Runoff
- Wave
More Information/Getting Help

CESM Bulletin Board: [http://bb.cgd.ucar.edu/](http://bb.cgd.ucar.edu/)

<table>
<thead>
<tr>
<th>Forum</th>
<th>Topics</th>
<th>Posts</th>
<th>Last-post</th>
</tr>
</thead>
<tbody>
<tr>
<td>Announcements</td>
<td>29</td>
<td>61</td>
<td>Invitation to participate in CESM data survey by alice. June 13, 2015 - 6:14pm</td>
</tr>
<tr>
<td>Bug reporting, Community Bug Reporting</td>
<td>194</td>
<td>625</td>
<td>CESM3 run error by janezhang8879@... July 21, 2015 - 3:03am</td>
</tr>
<tr>
<td>Climate Variability Diagnostics Package inquiries</td>
<td>2</td>
<td>20</td>
<td>Sign of PDO by argyll June 9, 2014 - 10:40am</td>
</tr>
<tr>
<td>General Discussion</td>
<td>434</td>
<td>1479</td>
<td>CLM4 irrigation Modification by mcfowler@... July 29, 2013 - 9:11am</td>
</tr>
<tr>
<td>This forum is for the discussion of git issues in the CIEME repository</td>
<td>3</td>
<td>16</td>
<td>sun external for a given git tag by andre May 6, 2015 - 4:04pm</td>
</tr>
<tr>
<td>Input Data inquiries</td>
<td>207</td>
<td>555</td>
<td>msg: v0.0.1.25_to_T85_assim_110411.nc by alice July 30, 2015 - 11:43am</td>
</tr>
<tr>
<td>Known Issues</td>
<td>0</td>
<td>0</td>
<td>n/a</td>
</tr>
<tr>
<td>Subforums: ocean/POP3 (0), atmosphere/CAM (3), atmosphere/WACOM (0), Component Sets (COMPSETs) (6), Coupler (3), Earth and Earth Systems Models (0), CCMs (0), cloud/CICE (2), land/OCLM (3), sisic/ESM (1), Machine/scripts (2), mapping (3), utilities (1)</td>
<td>0</td>
<td>0</td>
<td>n/a</td>
</tr>
<tr>
<td>CESM Intercomparison Project (MIP) inquiries</td>
<td>14</td>
<td>47</td>
<td>Notice to the Community: ESGF Nodes Going Offline by strandig June 21, 2015 - 10:36am</td>
</tr>
<tr>
<td>New Feature Requests</td>
<td>1</td>
<td>2</td>
<td>user_nit feature request by jedawen August 14, 2014 - 4:18pm</td>
</tr>
</tbody>
</table>
More Information/Getting Help

CESM tutorial: http://www.cesm.ucar.edu/events/tutorials/

Tutorials

Upcoming CESM Tutorials

- 2017 CESM Tutorial | 14 - 18 August 2017, National Center for Atmospheric Research, Mesa Lab, Boulder, CO

Past CESM Tutorials

- 2016 CESM Tutorial | 8 - 12 August 2016, NCAR, Mesa Lab, Boulder, CO
- 2016 CMM Tutorial | 16 - 18 August 2016, NCAR, Mesa Lab, Boulder, CO
- 2016 CLM Tutorial | 12 - 16 September 2016, NCAR, Mesa Lab, Boulder, CO
- 2015 CESM Tutorial | 8 - 14 August 2015, NCAR, Mesa Lab, Boulder, CO
- 2014 CESM Tutorial | 11 - 15 August 2014, NCAR, Mesa Lab, Boulder, CO
- 2014 CLM Tutorial | 18 - 21 February 2014, NCAR, Mesa Lab, Boulder, CO
- 2013 CESM Tutorial | 12 - 16 August 2013, NCAR, Boulder, CO
- 2012 CESM Tutorial | 30 July - 03 August 2012, NCAR, Boulder, CO
- 2011 CESM Tutorial | 1 - 5 August 2011, NCAR, Boulder, CO
- 2010 CESM Tutorial | 12 - 16 July 2010, NCAR, Boulder, CO
Thank You!

The UCAR Mission is:
To advance understanding of weather, climate, atmospheric composition and processes;
   To provide facility support to the wider community; and,
      To apply the results to benefit society.

NCAR is sponsored by the National Science Foundation
Day 1 Exercise 0

- This afternoon we will simply be introducing you to the system and running for the first time.
- Log in to cheyenne following the instructions on your compile card and follow these steps.

**Step 0:** If you are not familiar with the Linux csh environment, then review this cheat sheet with a list of common commands:

**Step 1:** From your tutorial machine window prompt, login to cheyenne:

```bash
ssh -Y cheyenne.ucar.edu
```

**One Time Setup:** Check your default login environment settings:

```bash
cp /glade/p/cesm/tutorial/tcshrc .tcshrc
source ~/.tcshrc
```

*NOTE:* all tutorial logins default to tcsh and environment settings are in ~/.tcshrc

**Step 2:** login to a cheyenne compute node.

```bash
cp /glade/p/cesm/tutorial/compile_node.csh ./compile_node.csh
./compile_node.csh
```

This will be the step that you will run just prior to running “case.build”

**Step 3:** check the queue status to see your session on the compute node

```bash
whoami
qstat -u [loginname]
```

**Step 4:** logout of cheyenne compute node.

```bash
logout
```

**Step 5:** logout of cheyenne compute node.

```bash
logout
```
Day 1 Exercise 1

• This afternoon we will simply be introducing you to the system and running for the first time. After each step, check the files in the CASEROOT, EXEROOT and DOUT_S_ROOT directories.
• Log in to cheyenne and run the following steps.

```bash
# One time step
mkdir ~/cases

# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm2_0_alpha07c/cime/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase --case ~/cases/b.day1.0 --res f19_g17 --compset B1850

# go into the case you just created in the last step
cd ~/cases/b.day1.0
./xmlquery CASEROOT

# (2) invoke case.setup
./case.setup
./xmlquery EXEROOT

# (3) build the executable on a cheyenne compute node
~/compile_node.csh
./case.build
Logout```
This afternoon we will simply be introducing you to the system and running for the first time. After each step, check the files in the CASEROOT, EXEROOT and DOUT_S_ROOT directories.

# (4) submit your run to the batch queue

NOTE – each day of the tutorial we will be using a different dedicated batch queue for submission. Prior to each case.submit command, you will want to run the following xmlchange commands:

Monday:
./xmlchange --subgroup case.run JOB_QUEUE=R1578614
./xmlchange --subgroup case.st_archive JOB_QUEUE=R1578614

Tuesday:
./xmlchange --subgroup case.run JOB_QUEUE=R1578615
./xmlchange --subgroup case.st_archive JOB_QUEUE=R1578615

Wednesday:
The dedicated queue name is R1585559 and will be used in the post-processing scripts.

Thursday:
./xmlchange --subgroup case.run JOB_QUEUE=R1578617
./xmlchange --subgroup case.st_archive JOB_QUEUE=R1578617

Fri: R1578619
./xmlchange --subgroup case.run JOB_QUEUE=R1578619
./xmlchange --subgroup case.st_archive JOB_QUEUE=R1578619

Now, submit
./case.submit
qstat -u [loginname]
./xmlquery DOUT_S_ROOT
Day 1 Exercises 2-3

# Exercise 1: Check on your case and resubmit when it is complete.
qstat -u [loginname]
cat CaseStatus

# Changing options like STOP_N and STOP_OPTION would increase run length.
../xmlchange CONTINUE_RUN=TRUE
../case.submit

# Note that if you make a mistake, you can kill the job using its ID number displayed when you run qstat
# qdel <job_id>

# Exercise 2: create_clone

# Go back to the scripts directory
cd /glade/p/cesm/tutorial/cesm2_0_alph07c/cime/scripts

# Make a clone of the case
./create_clone --clone ~/cases/b.day1.0 --case ~/cases/b.day1.2

# Take a look in the create_clone directory.

# What is the value of CONTINUE_RUN in the new directory (this is in env_run.xml)?
# What does README.case look like?
# What other files are copied over?
# What would be the next step in building and running the cloned case?
Notes for this tutorial

- There are a few things we will do this week that are different from running normally on cheyenne.
  - We will be using code in “/glade/p/cesm/tutorial” this week. Normally, you will use a version of the code in “/glade/p/cesm/releases”, or check out your own version. The tutorial code refers to a special account key that will not work in the future!
  - We will be taking turns building the model on a cheyenne compute node. Normally, you would build on a cheyenne login node and run on the batch nodes.
- Some general tips:
  - We will use short case directory names today, but in the future you may want to use longer names so that cases are easier to find. Typically, case names should include the compset, grid, and possibly a short name for the experiment.
  - While CESM is building, you can open a second terminal window and log in to cheyenne again. This allows you to look around or do other things while waiting for a job to complete.
Further exercises

Some suggestions if you finish early today:

- Look through the exercises from Christine Shields to get a preview of this Tuesday’s topics.

- Look through the CESM2.0 web page and other information online. Try to get a feel for what information you would need to look up to set up your own cases. [http://www.cesm.ucar.edu](http://www.cesm.ucar.edu) and [http://www.cesm.ucar.edu/models/cesm2.0](http://www.cesm.ucar.edu/models/cesm2.0) and [https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne](https://www2.cisl.ucar.edu/resources/computational-systems/cheyenne)

- Try using the “ncview” command on one of the history files in your run directory. This is a simple but useful tool for taking a quick look at output. First, look at the system modules loaded in your login environment:
  
  module list
  
  If ncview is not listed, then load it into your environment using:

  module load ncview

- Take a quick look at the NCO utilities for manipulating netCDF files: [http://nco.sourceforge.net/nco.html](http://nco.sourceforge.net/nco.html)

- PLEASE register as a new user on the DiscussCESM Forums website at: [http://bb.cgd.ucar.edu](http://bb.cgd.ucar.edu)
  
  Include a valid email, name, job title, and organization so I can approve your request and keep the spammers out!
Day 1 Auxiliary Exercises

In Wednesday’s lab session you will be learning how to run the various diagnostic packages. You will also learn about the types of tools that are commonly used on model output. Here are some exercises that you can do to prepare yourself for Wednesday’s lab session.

• Go to the CESM1 Large Ensemble Community Project page [http://www.cesm.ucar.edu/projects/community-projects/LENS/](http://www.cesm.ucar.edu/projects/community-projects/LENS/) After reading the project overview click on the “Diagnostics” link. Take a look at the available experiments and look at diagnostics output from the atmosphere, sea ice, land, and ocean diagnostics packages. Become familiar with the types of calculations the packages do.

• Go to each of the prognostic model web pages

• See [http://www.cesm.ucar.edu/working_groups/CVC/cvdp](http://www.cesm.ucar.edu/working_groups/CVC/cvdp). The Climate Variability Diagnostics Package (CVDP) is different from the other diagnostics packages in that it is usually run over an entire simulation and can be run on numerous simulations (CESM and non-CESM data) at once. The CVDP calculates the major modes of variability, trends, and provides a quantifiable metric table. Look at the website example comparisons.

• Go to [http://climatedataguide.ucar.edu](http://climatedataguide.ucar.edu) and explore the site. The Climate Data Guide contains information on over 150 different datasets, provides inter-dataset comparisons, and has dataset pros and cons evaluated by expert dataset users.

• The programming language NCL is used extensively within the CESM project. You will have the opportunity to run several NCL scripts on Wednesday. Take a look at the NCL Examples page to get an idea of the types of plots NCL can create: [http://www.ncl.ucar.edu/Applications/](http://www.ncl.ucar.edu/Applications/)