CESM Tutorial

NCAR Earth System Laboratory

CESM 1.2.x and CESM1.1.x
CESM1.0.x and previous (see earlier tutorials)

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
CESM Web Page Models
http://www2.cesm.ucar.edu/models/current

**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 Release Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>CESM 1.2.2</td>
</tr>
<tr>
<td>Release Notes</td>
</tr>
<tr>
<td>CESM 1.1.2</td>
</tr>
<tr>
<td>Notable Improvements</td>
</tr>
<tr>
<td>CESM 1.0.2</td>
</tr>
<tr>
<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.
Model Input data

Timing and load balance

Data management and distribution
Hardware/Software Requirements

- **Supported platforms**
  CESM currently runs "out of the box" today on the following machines
  - *yellowstone* – NCAR IBM
  - *edison* – NERSC Cray Cascade
  - *bluewaters* – NCSA Cray XE6
  - *mira* – ANL IBM Bluegene/Q
  - *pleiades* – NASA SGI ICE cluster
  - and a few others

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

- **Running CESM on other platforms**
  Require porting + software
  - Subversion client (version 1.4.2 or greater)
  - Fortran and C compilers (recommend pgi, intel, or ibm xlf compilers)
  - NetCDF library (recommend netcdf4.1.3 or later)
  - MPI (MPI1 is adequate, Open MPI or MPICH seem to work on Linux clusters)
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 cesm_setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
(A) Registration

- Go to CESM1.2 home page: http://www.cesm.ucar.edu/models/cesm1.2/

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

- Register -- you will be emailed a username and password
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 cesm_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-release.cgd.ucar.edu/model_versions)
  [https://svn-ccsm-models.cgd.ucar.edu/cesm1/release_tags](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`

Overview of Directories (after initial model download)

The CESM root consists of 2 directories: models and scripts

Code for all the components, coupler, etc…
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 cesm_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
  - 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 cesm_setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
(D) Porting

- On supported machines - no porting is necessary

- On new machines – porting needs to be done

Porting details are outside the scope of this tutorial

User’s Guide
Porting and Validating CESM on a new platform
Work Flow: Super Quick Start

CESM can be run with a set of 4 commands

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

1. # one time step
   mkdir ~/cases

2. # go into scripts directory into the source code download
   cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts

   # (1) create a new case in the directory “cases” in your home directory
   ./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

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

3. # (2) invoke cesm_setup
   ./cesm_setup

4. # (3) build the executable
   ./b.day1.0.build

   # (4) submit your run to the batch queue
   ./b.day1.0.submit

It is that easy! 😊
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 cesm_setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Overview of Directories
(+ before create_newcase)

CESM Download

~/cesm1_2_2
$CCSMROOT

models

scripts
create_newcase

This is the script you need to create a new case
Work Flow: Super Quick Start

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

```bash
# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

# (3) build the executable
./b.day1.0.build

# (4) submit your run to the batch queue
./b.day1.0.submit
```
(1) Create a new case

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

`create_newcase` requires 4 arguments

- What is the casename?
- Which resolution?
- Which model configuration?
- Which set of components?
- Which machine are you running on?
(1) `create_newcase` arguments

`create_newcase` requires 4 arguments

```
create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone
```
(1) create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

What is the casename?

case specifies the name and location of the case being created

~/cases/b.day1.0
(1) create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

Which resolution?

res specifies the model resolution (or grid)

New grid naming convention

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

Example of equivalent alias, short name and long name:
- alias: T31_g37 (atm/lnd_ocn/ice)
- short name: T31_gx3v7
- long name = a%T31_l%T31_oi%gx3v7_r%r05_m%gx3v7_g%null_w%null
(1) create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

Which 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, short name and long name. Example of equivalent alias, short name and long name:

- alias: B1850CN
- short name: B_1850_CN
- long name = 1850_CAM4_CLM40%CN_CICE_POP2_RTM_SGLC_SWAV
More on CESM component sets

Plug and play of components with different component models

Color code: active data stub

I_ DATM CLM
RTM cpl SGLC SWAV
SOCN SICE

G_
DATM SLND
DROF cpl SGLC SWAV
POP2 CICE

B_
CAM CLM
RTM cpl SGLC SWAV
POP2 CICE

F_
CAM CLM
RTM cpl SGLC SWAV
DOCN CICE (P)
(1) create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

Which machine are you running on?

mach specifies the machine that will be used.

“supported” machines tested regularly, eg. yellowstone, edison, mira, bluewaters
Valid Values for res, compset, and mach

Command line to list all the valid choices for grids, compsets and machines

```
./create_newcase -list <type>
```

with type can be [compsets, grids, machines]

List of valid values is also available from the CESM website

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

List of scientifically validated component sets and resolutions are available from the CESM website

https://www2.cesm.ucar.edu/models/scientifically-supported
About create_newcase

./create_newcase --help lists all the available options
Most often only four options are used: case, compset, res, and mach

cd ../cesm1_2_2.tutorial/scripts/
./create_newcase -help

SYNOPSIS
create_newcase [options]

OPTIONS
User supplied values are denoted in angle brackets (<>). Any value that contains white-space must be quoted. Long option names may be supplied with either single or double leading dashes. A consequence of this is that single letter options may NOT be bundled.

- case <name>         Specifies the case name (required).
- compset <name>      Specify a CESM compset (required).
- res <name>          Specify a CESM grid resolution (required).
- mach <name>         Specify a CESM machine (required).
- compiler <name>     Specify a compiler for the target machine (optional)
default: default compiler for the target machine
- mpilib <name>       Specify a mpi library for the target machine (optional)
default: default mpi library for the target machine allowed: openmpi, mpich, ibm, mpi-serial, etc redundant with _M confopts setting
- mach_dir <path>     Specify the locations of the Machines directory (optional).
default: /glade/p/cesm/cseg/collections/cesm1_2_0_beta08/scripts/ccsm_utils/Machines
- pecount <name>      Value of S,M,L,X1,X2 (optional).
default: M, partially redundant with confopts _P
- pes_file <name>     Full pathname of pes file to use (will overwrite default settings) (optional).
See sample_pes_file.xml for an example.
- user_compset Long name for new user compset file to use (optional)
This assumes that all of the compset settings in the long name have been defined.
- grid_file <name>    Full pathname of grid file to use (optional)
See sample_grid_file.xml for an example.
Note that compset components must support the new grid.
- help [or -h]        Print usage to STDOUT (optional).
- list <type>         Only list valid values, type can be [compsets, grids, machines] (optional).
...
Result of running create_newcase

```
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 ~/cases/b.day1.0/env_case.xml
```

For a list of potential issues in the current tag, please point your web browser to:
https://svn-ccsm-models.cgd.ucar.edu/cesm1/known_problems/

---

grid longname is T31_g37
Component set: longname (shortname) (alias)
1850_CAM4_CLM40%CN_CICE_POP2_RTM_SGLC_SWAV (B_1850_CN) (B1850CN)
Component set Description:
physics: clm4.0 cn specified phenology: prognostic cice: POP2 default:
Grid:
a%T31_l%T31_o1%gx3v7_r%r05_m%gx3v7_g%null_w%null (T31_gx3v7)
ATM_GRID = 48x96 NX_ATM=96 NY_ATM=48
LND_GRID = 48x96 NX_LND=96 NX_LND=48
...

Non-Default Options:
ATM_NCPL: 48
BUDGETS: TRUE
CAM_CONFIG_OPTS: -phys cam4
...

The PE layout for this case match these options:
GRID = a%T31.+oi%gx3
CCSM_LCOMPSET = CAM.+CLM.+CICE.+POP
MACH = Yellowstone
Creating /glade/u/home/hannay/cases/b.day1.0
Created /glade/u/home/hannay/cases/b.day1.0/env_case.xml
Created /glade/u/home/hannay/cases/b.day1.0/env_mach_pes.xml
Created /glade/u/home/hannay/cases/b.day1.0/env_build.xml
Created /glade/u/home/hannay/cases/b.day1.0/env_run.xml
Locking file /glade/u/home/hannay/cases/b.day1.0/env_case.xml
Successfully created the case for Yellowstone

Locking file ~/cases/b.day1.0/env_case.xml
Successfully created the case for Yellowstone

Success!
Overview of Directories (after create_newcase)

- **CESM Download**
  - ~/cesm1_2_2
  - $CCSMROOT
  - models
    - atm
    - Ind
    - ocn
    - ice
    - glc
    - drv
    - rof
    - wav
    - utils
    - csm_share

- **scripts**
  - create_newcase

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

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

- **LockedFiles**
  - Buildconf
  - LockedFiles
  - SourceMods
  - Tools

create_newcase creates case directory that contains:
- cesm_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
About env_*.xml files

- **env_*.xml** contains variables used by scripts -- some can be changed by the user
  - **env_case.xml**: set by create_newcase and cannot be modified
  - **env_mach_pes.xml**: specifies layout of components
  - **env_build.xml**: specifies build information
  - **env_run.xml**: sets run time information (such as length of run, frequency of restarts, …)
    
    User interacts with this file most frequently

- **Here’s a snippet of the env_run.xml file**

```xml
<entry id="STOP_OPTION" value="ndays" />

<entry id="STOP_N" value="5" />
```

“id” - variable name

“value” – variable value

- **To modify a variable in an xml file** – use `xmlchange`

  `xmlchange STOP_N=20`
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 cesm_setup
  (3) Build the Executable
  (4) Run the Model and Output Data Flow
Set of commands to build and run the model on a supported machine: ”yellowstone”

```bash
# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

# (3) build the executable
./b.day1.0.build

# (4) submit your run to the batch queue
./b.day1.0.submit
```
SYNOPSIS

* ./cesm_setup -help *

Creates Macros file for target machine if it does not exist
Creates user_nl_xxx files for target components
(and number of instances) if they do not exist
Creates batch run script (case.run) for target machine

USAGE

    cesm_setup [options]

OPTIONS

- help [or -h]          Print usage to STDOUT.
- clean                 Removes the batch run script for target machines
                         Macros and user_nl_xxx files are never removed
                         by cesm_setup - you must remove them manually
Calling cesm_setup

- `cd ~/cases/b.day1.0`
- `./cesm_setup`

Creating Macros file for yellowstone
/glade/p/cesm/cseg/tutorial/cesm1_2_2.tutorial/scripts/ccsm_utils/Machines/config_compilers.xml intel yellowstone
Creating batch script b.day1.0.run
Locking file env_mach_pes.xml
Creating user_nl_xxx files for components and cpl
Running preview_namelist script
infile is /glade/u/home/hannay/cases/b.day1.0/Buildconf/cplconf/cesm_namelist
CAM writing dry deposition namelist to drv_flds_in
CAM writing namelist to atm_in
CLM configure done.
CLM adding use_case 1850_control defaults for var sim_year with val 1850
CLM adding use_case 1850_control defaults for var sim_year_range with val constant
CLM adding use_case 1850_control defaults for var use_case_desc with val
Conditions to simulate 1850 land-use
CICE configure done.
POP2 build-namelist: ocn_grid is gx1v6
POP2 build-namelist: ocn_tracer_modules are iage
See ./CaseDoc for component namelists
If an old case build already exists, might want to run b.day1.0.clean_build before building
Overview of Directories (after cesm_setup)

- **CESM Download**
  - ~/cesm1_2_2
  - $CCSMROOT

- **CASE Directory**
  - ~/cases/b.day1.0
  - cesm_setup creates:
    - case scripts (to build, run and archive)
    - namelist modification files user_nl_***
      - this is where you modify your namelists
  - CASE Docs: contains copy of the namelists
  - CaseDocs: This is for reference only and files in this directory SHOULD NOT BE EDITED.
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 cesm_setup
  3. Build the Executable
  4. Run the Model and Output Data Flow
Set of commands to build and run the model on a supported machine: ”yellowstone”

# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

# (3) build the executable
./b.day1.0.build

# (4) submit your run to the batch queue
./b.day1.0.submit
Build the Model

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

- To completely rebuild, run *.clean_build* first

- The *.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 –export to acquire missing data
  - This will use svn to put required data in the inputdata directory
  - Then re-run build script
Running the .build Script

- Checks for missing input data
- Aborts if any input data is missing
- Builds the component model libraries and executable by running the *build.exe.csh* scripts for each component

```
./b.day1.0.build

CESM BUILDNML SCRIPT STARTING
- To prestage restarts, untar a restart.tar file into /glade/scratch/hannay/b.day1.0/run
  infile is /glade/u/home/hannay/cases/b.day1.0/Buildconf/cplconf/cesm_namelist
...
CESM BUILDNML SCRIPT HAS FINISHED SUCCESSFULLY

CESM PRESTAGE SCRIPT STARTING
- Case input data directory, DIN_LOC_ROOT, is /glade/p/cesm/cseg/inputdata
- Checking the existence of input datasets in DIN_LOC_ROOT
CESM PRESTAGE SCRIPT HAS FINISHED SUCCESSFULLY

CESM BUILDEXE SCRIPT STARTING
COMPILER is intel
- Build Libraries: mct gptl pio csm_share
Tue Jun 11 19:13:41 MDT 2013 /glade/scratch/hannay/b.day1.0/bld/mct/mct.bldlog.130611-191330
...
- Locking file env_build.xml
CESM BUILDEXE SCRIPT HAS FINISHED SUCCESSFULLY
```
Overview of Directories (after build)

CESM Download
~/cesm1_2_2
$CCSMROOT

- models
- scripts
  - create_newcase

CASE Directory
b.day1.0
 cesm_setup
 b.day1.0.build
 b.day1.0.submit
 user_nl_xxx

Buildconf
CaseDocs
LockedFiles
SourceMods
Tools

Build/Run Directory
/bglade/scratch/
userx/b.day1.0
$SEXEROOT

- bld
- run
  - $SRUNDIR

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

The build script
(1) checks input data
(2) creates a build/run directory with model executable and namelists

If any inputdata is missing,
- Build aborts and provides a list of missing files
- Run ./check_input_data --export 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
Set of commands to build and run the model on a supported machine: "yellowstone"

```
# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

# (3) build the executable
./b.day1.0.build

# (4) submit your run to the batch queue
./b.day1.0.submit
```
(4) Running the Model

When you submit your jobs

```bash
~cases/b.day1.0>b.day1.0.submit
c
check_case OK
Job <959733> is submitted to queue <regular>
```

Use "bjobs" to check if job is running

```bash
~cases/b.day1.0>bjobs
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>STAT</th>
<th>QUEUE</th>
<th>FROM_HOST</th>
<th>EXEC_HOST</th>
<th>JOB_NAME</th>
<th>SUBMIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>960463</td>
<td>userx</td>
<td>PEND</td>
<td>regular</td>
<td>yslogin3-ib</td>
<td></td>
<td>b.day1.0</td>
<td>Jun 17 08:34</td>
</tr>
</tbody>
</table>

Your job is waiting in the queue

```bash
~cases/b.day1.0>bjobs
```

<table>
<thead>
<tr>
<th>JOBID</th>
<th>USER</th>
<th>STAT</th>
<th>QUEUE</th>
<th>FROM_HOST</th>
<th>EXEC_HOST</th>
<th>JOB_NAME</th>
<th>SUBMIT_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>960463</td>
<td>userx</td>
<td>RUN</td>
<td>regular</td>
<td>yslogin3-ib</td>
<td>15*ys0702-ib</td>
<td>b.day1.0</td>
<td>Jun 17 08:34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys1872-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys1906-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys1907-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys1908-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys1918-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2055-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2057-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2058-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2130-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2131-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2132-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2216-ib</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15*ys2218-ib</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Your job is running
When running, the model scripts write files into your run directory.

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

CESM Download
~/cesm1_2_2
$CCSMROOT

models
scripts
create_newcase

atm
Ind
ocn
ice
glc
drv
rof
wav
utils
csm_share

CASE Directory
b.day1.0
cesm_setup
b.day1.0.build
b.day1.0.submit
user_nl_xxx

Buildconf
CaseDocs
Tools
Source Mods
Timing
Logs

Build/Run Directory
/glade/scratch/
userx/b.day1.0
$EXEROOT

bld
run
$RUNDIR

Buildconf

Short Term Archive
/glade/scratch/
archive/b.day1.0
$DOUT_S_ROOT

atm
Ind
ocn
ice
glc
rof
cpl
wav
lib

Short Term Archive

HPSS
/rest
dart
-lib
wav
rest

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

share
cpl
atm
Ind
ocn
ice
glc
wav
rof

(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
(4) Move data to permanent long-term storage
Set DOUT_L to TRUE in env_run.xml
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 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/modelsCESM1.2/
More Information/Getting Help

CESM Bulletin Board: http://bb.cgd.ucar.edu/
More Information/Getting Help

CESM tutorial: http://www.cesm.ucar.edu/events/tutorials/
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 yellowstone, geyser or caldera depending on the instructions on your compile card and follow these steps.

Step 1: From your tutorial machine window prompt, type the first command from your compile card. This creates a login session on yellowstone.

If your compile card is blue and says “Yellowstone” at the top then congratulations! You are done logging in and have successfully completed this first exercise.

If your compile card is yellow or green and says “Geyser Login” or “Caldera Login” at the top then you need to type one additional command – but just for today!

Step 2 (one time only – this step is not on your compile card):
```
# cp /glade/p/cesm/tutorial/login_scripts/* .csh
```

Step 3:
Type in the last command on your compile card
```
# ./caldera.csh calderaXX
-- or –
# ./geyser.csh geyserXX
```
This afternoon we will simply be introducing you to the system and running for the first time.

Log in to yellowstone, geyser or caldera and run the following steps.

Do the build step only if you have a compile card.

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

# go into scripts directory into the source code download
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/scripts

# (1) create a new case in the directory “cases” in your home directory
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B1850CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

# (3) build the executable
./b.day1.0.build

# (4) submit your run to the batch queue
./b.day1.0.submit
```
Day 1 Exercises 2-3

# Exercise 1: Check on your case and resubmit when it is complete.
bjobs
cat cesm.stdout.*

# Changing options like STOP_N and STOP_OPTION would increase run length.
./xmlchange CONTINUE_RUN=TRUE
./b.day1.0.submit

# Note that if you make a mistake, you can kill the job using its ID number.
# bkill <job_id>

# Exercise 2: create_clone

# Go back to the scripts directory
cd /glade/p/cesm/tutorial/cesm1_2_2.tutorial/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?
There are a few things we will do this week that are different from running normally on yellowstone.

- We will be using code in “/glade/p/cesm/tutorials” this week. Normally, you will use a version of the code in “/glade/p/cesm/collections”, or check out your own version. *The tutorial code refers to a special account key that will not work in the future!*
- Some of you will be building on caldera or geyser today. Normally, you would build on the yellowstone login nodes and run on the batch nodes, and use caldera or geyser only for data postprocessing and analysis.

**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 yellowstone 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 attached exercises from Adam Phillips to get a preview of this Wednesday’s topics.
  • Look through the user’s guide and other information online. Try to get a feel for what information you would need to look up to set up your own cases.
  • 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.
  • Take a quick look at the NCO utilities for manipulating netCDF files: http://nco.sourceforge.net/nco.html
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://www2.cesm.ucar.edu/models/experiments/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.

• See http://www2.cesm.ucar.edu/working-groups/cvcwg/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 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/