CESM Basic Workflow

Cécile Hannay

NCAR is sponsored by the National Science Foundation
Outline

• CESM at a glance
  1) The CESM framework
  2) Overview of CESM directories
  3) The CESM webpage

• CESM workflow
  1) Create a new case
  2) Invoke cesm_setup
  3) Build the executable
  4) Run and output data

• Getting help

• Appendix
The CESM framework

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

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

The CESM can be run through a set of scripts provided with the model.

This talk is a quick start to the CESM workflow (out-of-the-box)

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

CESM lives in 2 directories:

- CESM code
  /glade/p/cesm/lmwg/CLM2014_tutorial_n03_clm4_5_62
  $CCSMROOT

- CESM data
  /glade/p/cesm/cseg/inputdata
Overview of CESM Directories

CESM Code

/isode/p/cesm/lmgw/
CLM2014_tutorial_n03_clm4_5_62
$CCSMROOT

- models: contains the code for every component

- scripts: contains the scripts you need to run CESM

CESM data

/glade/p/cesm/cseg/inputdata
$DIN_LOC_ROOT
Overview of CESM Directories

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

If you use your own machine (not for this tutorial):

- 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 (i.e., by using `svn co`)
CESM 1.2 Web Page
http://www.cesm.ucar.edu/models/cesm1.2/

**CESM Models**

**Release Notes**

*Scientific validation*

*Guidance on model versions*

*Post processing Tools*

*Model Documentation*

**Background and Sponsors**

**How to acquire the code**

**Reporting problems**

**Getting Help**
CESM 1.2 Web Page
http://www.cesm.ucar.edu/models/cesm1.2/

Model Input data
Timing and load balance

Data management and distribution

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

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

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

CESM SUPPORT POLICY
CESM Support Policy - November 2012

CESM DATA MANAGEMENT & DISTRIBUTION PLAN
The annual CESM Support Plan (draft) documents the procedures for the storage and distribution of data associated with the CESM project.

Model Input data
Timing and load balance

EXTERNAL LIBRARY DOCUMENTATION
- Parallel I/O Library (PO)
- Model Coupling Toolkit (MCTI)
- Earth System Modeling Framework (ESMF)

MODEL INPUT DATA
The input data necessary to run all supported component sets is made available from a public Subversion input data repository. Note that the inputdata repository has much more data in it than you need to run CESM1.2 — DO NOT attempt to svn checkout the whole input data repository. The CESM1.2 User’s Guide explains how to obtain the subset of input data required for your needs.

PERFORMANCE AND LOAD BALANCING DATA
The timing table provides performance data that will continue to evolve due to changes in the model, machine hardware and input from the user community. For CESM1.2, please refer to the CESM1.1 Timing Table.
Outline

• CESM
  1) The CESM framework
  2) Overview of CESM directories
  3) CESM webpage

• CESM workflow
  1) Create a new Case
  2) Invoke cesm_setup
  3) Build the executable
  4) Run and output data

• Getting More Help

• Appendix
**Work Flow: Super Quick Start**

**CESM can be run with a set of 4 commands**

1. Go into scripts directory into the source code download
   
   ```bash
   cd /glade/p/cesm/lmwg/CLM2014_tutorial_n03_clm4_5_62/scripts
   ```

2. (1) Create a new case in your home directory
   
   ```bash
   ./create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone
   ```

3. Go into the case you just created in the last step
   
   ```bash
   cd ~/I1850_001
   ```

4. (2) Invoke cesm_setup
   
   ```bash
   ./cesm_setup
   ```

5. (3) Build the executable
   
   ```bash
   ./I1850_001.build
   ```

6. (4) Submit your run to the batch queue
   
   ```bash
   ./I1850_001.submit
   ```

It is that easy!
Basic Work Flow
(or how to set up and run an experiment)

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 CESM directories (+ before create_newcase)

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

CESM can be run with a set of 4 commands

1. # go into scripts directory into the source code download
cd /glade/p/cesm/lmwg/CLM2014_tutorial_n03_clm4_5_62/scripts

   # (1) create a new case in your home directory
   ./create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone

2. # go into the case you just created in the last step
   cd ~/I1850_001

   # (2) invoke cesm_setup
   ./cesm_setup

3. # (3) build the executable
   ./I1850_001.build

4. # (4) submit your run to the batch queue
   ./I1850_001.submit

It is that easy!
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?

create_newcase -case I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone
create_newcase converts arguments

create_newcase requires 4 arguments

create_newcase -case ~/l1850_001 -res f19_g16 -compset l1850CLM45 -mach yellowstone
create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/I1850_001 -res f19_g16 -compset l1850CLM45 -mach yellowstone

dispensable is the name and location of the case being created
~/I1850_001

Reommendation: Use meaningful names
create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone

res specifies the **model resolutions** (or grid): f19_g16 (atm/lnd_ocn/ice)

**Grid naming convention**

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

Example of equivalent alias, short name and long name:
- alias: f19_g16 (atm/lnd_ocn/ice)
- short name: 1.9x2.5_gx1v6
- long name = a%1.9x2.5_l%1.9x2.5_oi%gx1v6_r%r05_m%gx1v6_g%null_w%null
  atm  Ind  ocn/ice  river  Ind  Ind-ice  wave
  mask
create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone

compset specifies the "component set"

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

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: I1850CLM45
- short name: I_1850_CLM45
- long name = 1850_DATM%QIA_CLM45%SP_SICE_SOCN_RTM_SGLC_SWAV
More on CESM component sets

Plug and play of components with different component models

Color code: active data stub

Plug and play

B_ CAM CLM
RTM POP2
CICE

F_ CAM CLM
RTM DOCN
CICE (P)
create_newcase arguments

create_newcase requires 4 arguments

create_newcase -case ~/I1850_001 -res f19_g16 -compset l1850CLM45 -mach yellowstone

mach specifies the machine that will be used.

“supported” machines tested regularly, eg. yellowstone, titan, hopper, intrepid
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/
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_0/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.
- help [or -h]        Print usage to STDOUT (optional).
- list <type>         Only list valid values, type can be [compsets, grids, machines] (optional).

...
Overview of Directories
(after create_newcase)

**CASE Directory**

- ~/I1850_001
- $CASEROOT
- cesm_setup
- env_*xml
- xmlchange

**CASE Directory**

create_newcase creates case directory that contains:

- CESM Code
  - ~/cesm1_2_0
  - $CCSMROOT
  - models
  - scripts
    - create_newcase
  - atm
  - lnd
  - ocn
  - ice
  - glc
  - drv
  - rof
  - wav
  - utils
  - csm_share

**INPUTDATA Directory**

/glade/p/cesm/cseg/inputdata

$DIN_LOC_ROOT

**INPUTDATA Directory**

share cpl atm lnd ocn ice glc wav rof

**INPUTDATA Directory**

LockedFiles

SourceMods

Tools

Buildconf

**cesm_setup**: script used in the next step
files with xml variables used by CESM

**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

```
<!--"sets the run length in conjunction with STOP_N and STOP_DATE, valid values: none,never,nstep,nseconds,nsecond,nminutes,nminute,nhours,nhour,ndays,nday,nmonths,nmonth,nyears,nyear,date,ifdays0,end (char) " -->
<entry id="STOP_OPTION" value="ndays" />

<!--"sets the run length in conjunction with STOP_OPTION and STOP_DATE (integer) " -->
<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

CESM will run for 5 days
Basic Work Flow
(or how to set up and run an experiment)

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**

CESM can be run with a set of 4 commands

1. **# (1) create a new case in your home directory**
   
   ```
   ./create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone
   ```

2. **# go into the case you just created in the last step**
   
   ```
   cd ~/I1850_001
   ```

3. **# (2) invoke cesm_setup**

4. **# (3) build the executable**

   ```
   ./I1850_001.build
   ```

5. **# (4) submit your run to the batch queue**

   ```
   ./I1850_001.submit
   ```
(2) Overview of Directories (after cesm_setup)

CESM Code

~/cesm1_2_0 $CCSMROOT

models

scripts

create_newcase

atm

Ind

ocn

ice

glc

drv

rof

wav

utils

csm_share

CASE Directory

~/I1850_001

cesm_setup

I1850_001.build

I1850_001.submit

user_nl_xxx

CaseDocs

LockedFiles

SourceMods

Tools

cesm_setup creates:

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

CaseDocs: contains copy of the namelists
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)

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

CESM can be run with a set of 4 commands

# go into scripts directory into the source code download
cd /glade/p/cesm/lmwg/CLM2014_tutorial_n03_clm4_5_62/scripts

(1) # (1) create a new case in your home directory
./create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone

# go into the case you just created in the last step
cd ~/I1850_001

(2) # (2) invoke cesm_setup
./cesm_setup

(3) # (3) build the executable
./I1850_001.build

(4) # (4) submit your run to the batch queue
./I1850_001.submit
(3) Overview of Directories (after build)

If any input data 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

The build script checks input data. If any input data 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)

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

**CESM can be run with a set of 4 commands**

<table>
<thead>
<tr>
<th>Step</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>cd /glade/p/cesm/lmwg/CLM2014_tutorial_n03_clm4_5_62/scripts&lt;br&gt;create_newcase -case ~/I1850_001 -res f19_g16 -compset I1850CLM45 -mach yellowstone</td>
</tr>
<tr>
<td>2.</td>
<td>cesm_setup</td>
</tr>
<tr>
<td>3.</td>
<td>I1850_001.build</td>
</tr>
<tr>
<td>4.</td>
<td>I1850_001.submit</td>
</tr>
</tbody>
</table>
(4) Running the Model

When you submit your jobs

```
~I1850_001$I1850_001.submit
check_case OK
Job <959733> is submitted to queue <regular>
```

Use “bjobs” to check if job is running

```
~/I1850_001>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>I1850_001</td>
<td>Jun 17 08:34</td>
</tr>
</tbody>
</table>
```

Your job is waiting in the queue

```
~/I1850_001>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-i</td>
<td>I1850_001</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)

(4) 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_S to TRUE in env_run.xml (not for this tutorial)
Outline

• CESM
  1) The CESM framework
  2) Overview of CESM directories
  3) CESM webpage

• CESM workflow
  1) Create a new Case
  2) Invoke cesm_setup
  3) Build the executable
  4) Run and output data

• Getting More Help

• Appendix
More Information/Getting Help

Model User Guides: http://www.cesm.ucar.edu/models/cesm1.2/
More Information/Getting Help

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

- Registration
- Download the source code
- Hardware/software requirements
(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
(B) Download the Source Code

• Code and input datasets are in a subversion repository (*)
  https://svn-ccsm-release.cgd.ucar.edu/model_versions

• List the versions available on the CESM repository
  svn list https://svn-ccsm-release.cgd.ucar.edu/model_versions

• Check out a working copy from the repository (“Download code”)
  svn co https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_2_0

(*) You can get subversion at http://subversion.apache.org/
(C) Hardware/Software Requirements

- **Supported platforms**
  CESM currently runs “out of the box” today on the following machines
  - yellowstone – NCAR IBM
  - titan – ORNL Cray XK6
  - hopper – NERSC CrayXE6
  - edison – NERSC Cray Cascade
  - bluewaters – ORNL CrayXE6
  - intrepid – ANL IBM BlueGene/P
  - mira – ANL IBM BlueGene/Q
  - janus – Univ Colorado HPC cluster
  - pleiades – NASA SGI ICE cluster
  - and a few others

- **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)