

CESM Tutorial

**NCAR Earth System Laboratory
CESM Software Engineering Group**

CCSM 4.0 (released April 2010)

CESM 1.0 (released June 2010)

.....

CESM 1.0.4 (released Feb 2012)

NCAR is sponsored by the National Science Foundation



Outline

- **Release Homepage on Web**
- **Software & Hardware Requirements**
- **One-Time Setup**
 - A) **Registration and Source Code Download**
 - B) **Create an Input Data Root Directory**
 - C) **Porting**
- **Creating & Running a Case**
 - 1) **Create a New Case**
 - 2) **Configure the Case**
 - 3) **Build the Executable**
 - 4) **Initial Run and Output Data**
 - 5) **Continuation Runs**
- **Getting More Help**
- **Appendix**

CESM 1.0 Release Web Page

<http://www.cesm.ucar.edu/models/cesm1.0/>

The screenshot shows the CESM 1.0 Release Web Page with the following sections and content:

- Community Earth System Model** (Header)
- ABOUT CESM 1.0**
 - The Community Earth System Model (CESM) is a coupled climate model for simulating the earth's climate system. Composed of four separate models simultaneously simulating the earth's atmosphere, ocean, land surface and sea-ice, and one central coupler component, the CESM allows researchers to conduct fundamental research into the earth's past, present and future climate states.
 - Notable Model Improvements
 - CESM1.0 Release Notes (summaries of all releases in the CESM1.0 series)
 - Scientifically Supported Model Configurations
 - CESM Configuration Naming Conventions
- MODEL OUTPUT DATA AND DIAGNOSTICS**
 - Experiments & Diagnostics
 - Model Output Data (ESQ)
 - Post Processing Utilities
 - Model File Naming Conventions
 - Experiment Case Naming Conventions
- MODEL DOCUMENTATION**
 - CESM1.0**
 - User's Guide
 - Atmosphere Models**
 - Community Atmosphere Model (CAM5)
 - Climatological Data Model (DATM)
 - Land Models**
 - Community Land Model (CLM4)
 - Climatological Data Model (DLND)
 - Sea Ice Models**
 - Community Ice Code (CICE4)
 - Climatological Ice Model (DICE)
 - Ocean Models**
 - Parallel Ocean Program (POP2)
 - Climatological/Slab-Ocean Data Model (DOCN)
 - Land Ice Models**
 - Community Ice Sheet Model (Glimmer - CISM)
 - CESM Coupler**
 - CESM Coupler (CPL7)
- External Library Documentation**
 - Parallel I/O Library (PIO)
 - Model Coupling Toolkit (MCT)
 - Earth System Modeling Framework (ESMF)
- MODEL INPUT DATA**
 - The input data necessary to run all supported component sets is made available from a public Subversion input data repository. Note that the inputdata repository has much more data in it than you need to run CESM1.0 ---- **DO NOT attempt to svn checkout the whole input data repository**. The CESM1.0 User's Guide explains how to obtain the subset of input data required for your needs.
- PERFORMANCE AND LOAD BALANCING DATA**
 - The CESM1 Timing Table provides performance data that will continue to evolve due to changes in the model, machine hardware and input from the user community.
- CESM PROJECT**
 - The Community Earth System Model (CESM) is a fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.
 - CESM is sponsored by the National Science Foundation (NSF) and the U.S. Department of Energy (DOE). Administration of the CESM is maintained by the Climate and Global Dynamics Division (CGD) at the National Center for Atmospheric Research (NCAR).
- MODEL SOURCE CODE**
 - Copyright and Terms of Use**
 - All CESM source code is subject to the following Copyright Notice and Disclaimer.
 - Acquiring the Release Code**
 - CESM source code is distributed through a public Subversion code repository. This code can be checked out using Subversion client software, such as the command tool svn, or simply view the latest version with a web browser.
 - 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 CESM1.0 User's Guide.
 - Version Summaries and Release Notes**
 - The CESM1.0 Release Notes page summarizes all available versions of CESM1.0, along with details of new features, bug fixes, and any known issues identified in each release.
 - Reporting a Problem**
 - If you have any problems, please first read the User's Guide including the sections on FAQs and Use Cases. Please also refer to the CESM Bulletin Board, which is in place to facilitate communication within the CESM community. Finally, please also refer to the Release Notes entries that are provided with every release and release update. If questions or problems still exist, then please send an email to cesm-help@cgd.ucar.edu. Support questions will be answered as resources are available.

Notable Improvements



Data, Diagnostics, Post-Processing Tools



User's Guide



Component Model Documentation



External Libraries



Input Data



Timing Table



Background and Sponsors



Copyright and Terms of Use



How to Acquire the Code



Version Summaries



Reporting Problems, Known Problems



Software/Hardware Requirements

- 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)
 - [Note: Other external libraries (ESMF, MCT, PIO) are included in CESM source code, and do not have to be separately installed.]
-
- CESM currently runs “out of the box” today on the following machines
 - **bluefire** – NCAR IBM AIX
 - **jaguarpf (titan)** – ORNL Cray XT6
 - **hopper** – NERSC Cray XE6
 - **kraken** – NICS Cray XT5
 - **intrepid** – ANL IBM Bluegene/P
 - **janus** – Univ Colorado HPC cluster
 - **pleiades** – NASA SGI ICE cluster
 -
 - **and a few others**

Basic Work Flow

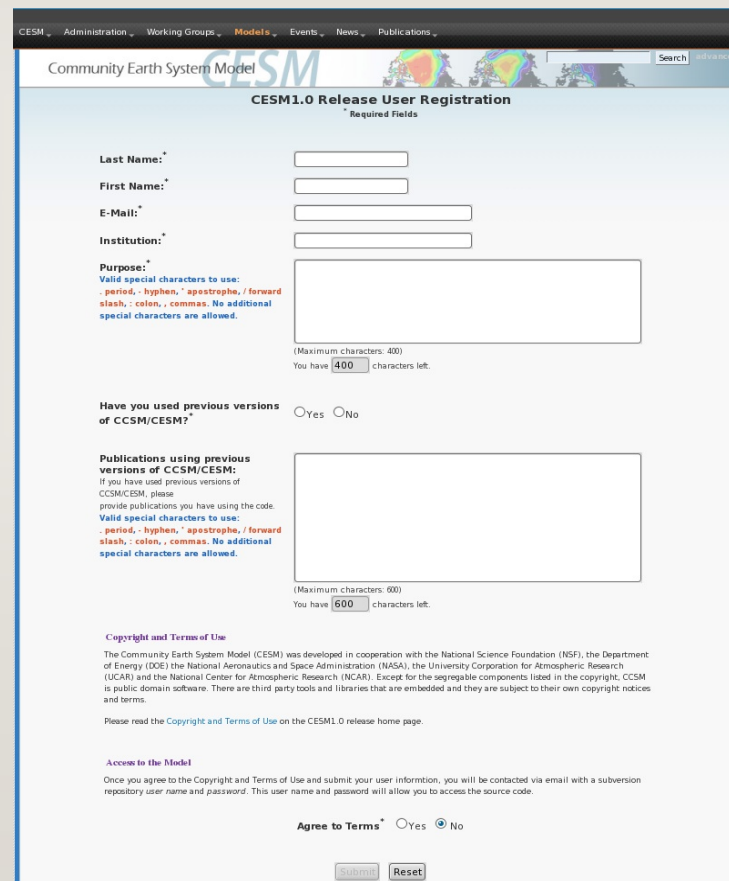
(or how to set up and run an experiment)

- **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) **Configure the Case**
 - (3) **Build the Executable**
 - (4) **Run the Model: Initial Run and Output Data Flow**
 - (5) **Run the Model: Continuation Run(s)**

(A) Registration

- Go to CESM1.0 home page:
- <http://www.cesm.ucar.edu/models/cesm1.0/>
- Right hand column has a link to the registration page, click on it
- Register -- you will be emailed a username and password



The screenshot shows the 'CESM1.0 Release User Registration' page. At the top, there is a navigation menu with links for 'Administration', 'Working Groups', 'Models', 'Events', 'News', and 'Publications'. Below the menu is a search bar and the 'CESM' logo. The main heading is 'CESM1.0 Release User Registration' with a note that asterisks indicate required fields. The form includes input fields for 'Last Name', 'First Name', 'E-Mail', and 'Institution'. A 'Purpose' section has a text area with a character count of 400. A question 'Have you used previous versions of CCSM/CESM?' has radio buttons for 'Yes' and 'No'. Another 'Publications using previous versions of CCSM/CESM' section has a text area with a character count of 600. At the bottom, there is a 'Copyright and Terms of Use' section with a paragraph of text and a 'Please read the Copyright and Terms of Use on the CESM1.0 release home page.' instruction. Below that is an 'Access to the Model' section with a paragraph of text. At the very bottom, there is an 'Agree to Terms' section with radio buttons for 'Yes' and 'No', and 'Submit' and 'Reset' buttons.

(A) Download the Source Code

- Code and input datasets are in a subversion repository
- Get subversion at <http://subversion.apache.org/>
- You need to download source code – but scripts will automatically download input data

```
svn list --username guestuser
```

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

```
>svn list --username guestuser https://svn-ccsm-release.cgd.ucar.edu/model_versions
Error validating server certificate for 'https://svn-ccsm-release.cgd.ucar.edu:443':
- The certificate is not issued by a trusted authority. Use the
  fingerprint to validate the certificate manually!
- The certificate hostname does not match.
- The certificate has expired.
Certificate information:
- Hostname: localhost.localdomain
- Valid: from Wed, 20 Feb 2008 23:32:25 GMT until Thu, 19 Feb 2009 23:32:25 GMT
- Issuer: SomeOrganizationalUnit, SomeOrganization, SomeCity, SomeState, --
- Fingerprint: 86:01:bb:a4:4a:e8:4d:8b:e1:f1:01:dc:60:b9:96:22:67:a4:49:ff
(R)eject, accept (t)emporarily or accept (p)ermanently? p
Authentication realm: <https://svn-ccsm-release.cgd.ucar.edu:443> ccsm:release
Password for 'guestuser': *****
ccsm4_0/
cesm1_0/
```

login

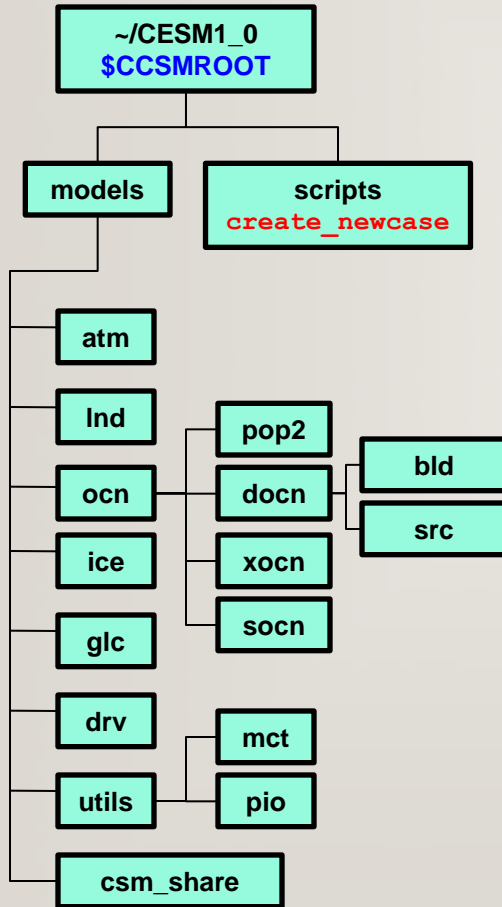
password

```
svn co --username guestuser
```

```
https://svn-ccsm-release.cgd.ucar.edu/models\_versions/cesm1\_0\_4
```

(A) Overview of Directories (initial model download)

CCSM Download



Basic Work Flow

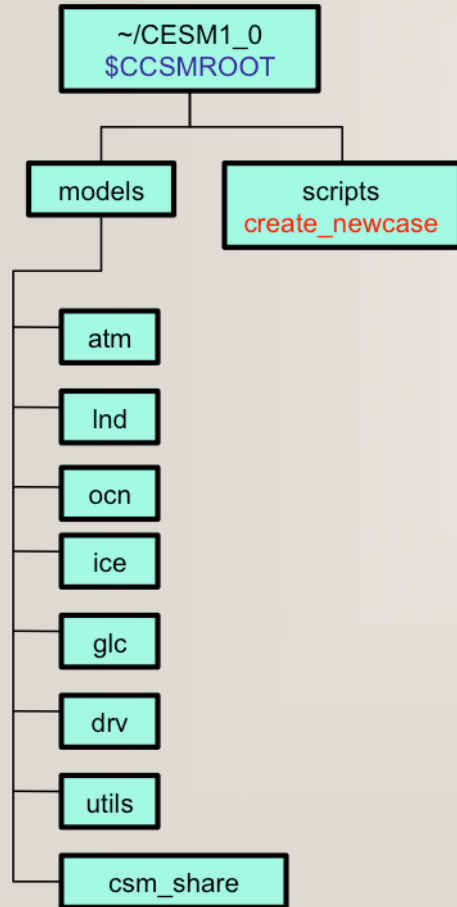
- **One-Time Setup Steps**
 - (A) Registration and Download
 - (B) **Create an Input Data Root Directory**
 - (C) Porting
- **Creating & Running a Case**
 - (1) Create a New Case
 - (2) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

(B) Create an Inputdata Root Directory

- The inputdata area contains all input data required to run the model
- Location specified in the scripts by the `$DIN_LOC_ROOT_CSMDATA` variable in file `env_run.xml`
- **On supported machines** - populated inputdata directory already exists
- **On non-supported machines** - need to create inputdata root directory
- Ideally directory is shared by a group of users to save disc space
- Initially inputdata directory is empty – data is added on an as-needed basis
- ***Do NOT download input data manually*** (ie. by using `svn co`)
- The script `check_input_data` is used to download input data
- Checks if necessary data is available in inputdata directory
- Downloads ***only*** the data needed for a particular run (more later)
- Puts the data in the proper subdirectories of the input data directory tree and creates the proper subdirectories if necessary
- ***Do NOT download input data manually***

(B) Overview of Directories (+ inputdata directory)

CESM Download



Basic Work Flow

- **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) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

(C) Porting

- Porting details are outside scope of tutorial –see User’s Guide on web and tutorial Appendix
- **On supported machines** - no porting is necessary
- **On new machines** – porting will need to be done
- **If the new machine is similar to a supported machine then porting can be relatively easy**
- **Porting might also be more challenging – a lot depends on the specifics of your machine**
- **See User’s Guide**

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine: "bluefire"

```
# go to root directory of source code download
cd /path/to/source/code/download/cesm1_0/

# go into scripts subdir
cd scripts

# (1) create a new case in your home dir
./create_newcase -case ~/mycase.01 -res f19_g16 -compset B_1850 -mach bluefire

# go into the case you just created in the last step
cd ~/mycase.01/

# (2) configure the case
./configure -case

# (3) build the executable
./mycase.01.bluefire.build

# (4) submit an initial run to the batch queue
bsub < mycase.01.bluefire.run

# check status of job and output files
bjobs
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# when the initial run finishes, change to a continuation run
./xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE

# (5) submit a continuation run to the batch queue
bsub < mycase.01.bluefire.run

# check status of job and output files
bjobs
ls -lFt $RUNDIR
ls -l logs
```

Basic Work Flow

- **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) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine named "bluefire"

```
# go to root directory of source code download
cd /path/to/source/code/download/cesm1_0/

# go into scripts subdir
cd scripts

# (1) create a new case in your home dir
./create_newcase -case ~/mycase.01 -res f19_g16 -compset B_1850 -mach bluefire

# go into the case you just created in the last step
cd ~/mycase.01/

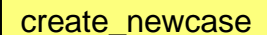
# (2) configure the case
./configure -case

.....
```


(1) Create a New Case

- Go to the scripts directory: `.../CESM1_0_4/scripts/`
- `create_newcase` is the tool that generates a new case
- Scripts are a combination of `csh`, `perl`, `sh`, and `xml`
- First step – run `create_newcase`

```
CESM1_0/scripts>ls -l
total 400
-rw-r--r--    1 userx   ncar      18596 May 12 11:33 ChangeLog
-rw-r--r--    1 userx   ncar         168 May 12 11:33 README
-rw-r--r--    1 userx   ncar         103 May 12 11:33 SVN_EXTERNAL_DIRECTORIES
drwxr-xr-x   10 userx   ncar      8192 May 12 11:33 ccsml_utils
-rwxr-xr-x    1 userx   ncar     19039 May 12 11:33 create_clone
-rwxr-xr-x    1 userx   ncar    52338 May 12 11:33 create_newcase
-rwxr-xr-x    1 userx   ncar    18253 May 12 11:33 create_test
-rwxr-xr-x    1 userx   ncar     9643 May 12 11:33 create_test_suite
drwxr-xr-x    3 userx   ncar      8192 May 12 11:33 doc
-rwxr-xr-x    1 userx   ncar     1255 May 12 11:33 link_dirtree
-rw-r--r--    1 userx   ncar         295 May 12 11:33 sample_compset_file.xml
-rw-r--r--    1 userx   ncar         851 May 12 11:33 sample_pes_file.xml
```



create_newcase

(1) About `create_newcase`

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


```
CESM1_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.



| | |
|---|--|
| <code>-case <name></code> | Specifies the case name (required). |
| <code>-compset <name></code> | Specify a CESM compset (required). |
| <code>-res <name></code> | Specify a CCSM grid resolution (required). |
| <code>-mach <name></code> | Specify a CESM machine (required). |
| <code>-pecount <name></code> | Value of S,M,L,X1,X2 (optional). (default is M). |
| <code>-pes_file <name></code> | Full pathname of pes setup file to use (will overwrite default setting). |
| <code>-compset_file <name></code> | Full pathname of compset setup file to use. (optional) |
| <code>-help [or -h]</code> | Print usage to STDOUT (optional). |
| <code>-list</code> | Only list valid values for compset, grid settings and machines (optional). |
| <code>-silent [or -s]</code> | Turns on silent mode - only fatal messages issued (optional). |
| <code>-verbose [or -v]</code> | Turn on verbose echoing of settings made by create_newcase (optional). |
| <code>-xmlmode <name></code> | Sets format of xml files; normal or expert (optional). (default is normal) |

The following arguments are required for a generic machine. Otherwise, they will be ignored.

| | |
|---|---|
| <code>-scratchroot <name></code> | ccsm executable directory (EXEROOT will be scratchroot/CASE) (char) |
| <code>-din_loc_root_csmdata <name></code> | cesm input data root directory (char) |
| <code>-max_tasks_per_node <value></code> | maximum mpi tasks per machine node (integer) |

The following two arguments turn on single point mode.

If one is given -- both MUST be given.

| | |
|-------------------------------------|--|
| <code>-pts_lat <value></code> | Latitude of single point to operate on (optional) |
| <code>-pts_lon <value></code> | Longitude of single point to operate on (optional) |

(1) `create_newcase`: Four Required Arguments

```
./create_newcase -case ~/cases/mycase1 -res f19_g16 -compset B_1850  
-mach bluefire
```

- “case” is the name and location of the case being created
 - `~/cases/mycase1`
- “res” specifies the model resolutions (or grid)
 - Format is `[atm/lnd grid]_[ocn/ice grid]`
 - Equivalent short and long names (`f19_g16 == 1.9x2.5_gx1v6`)
- “compset” specifies the “component set”
 - component set specifies component models, forcing scenarios and physics options for those models
 - Equivalent short and long names (`B1850CN == B_1850_CN`)
- “mach” specifies the machine that will be used.
 - “supported” machines tested regularly, eg. `bluefire`, `jaguar`, `franklin`, `intrepid`
 - “prototype” machines are not tested regularly, eg. `prototype_frost`
 - “generic machines” provide a starting point for porting, eg. `generic_ibm`
- values set on the command line are “locked down” in case directory
 - file `env_case.xml` contains these “locked down” variables

(1) Valid Values for res, compset, and mach

- `./create_newcase -list`

- lists all the valid choices for these command line options

```
CESM1_0/scripts>./create_newcase -list
```

RESOLUTIONS: name (shortname)

```
0.9x1.25_0.9x1.25 (f09_f09)
0.9x1.25_gx1v6 (f09_g16)
1.9x2.5_1.9x2.5 (f19_f19)
1.9x2.5_gx1v6 (f19_g16)
T31_gx3v7 (T31_g37)
ne30np4_1.9x2.5_gx1v6 (ne30_f19_g16)
```

COMPSETS: name (shortname): description (status)

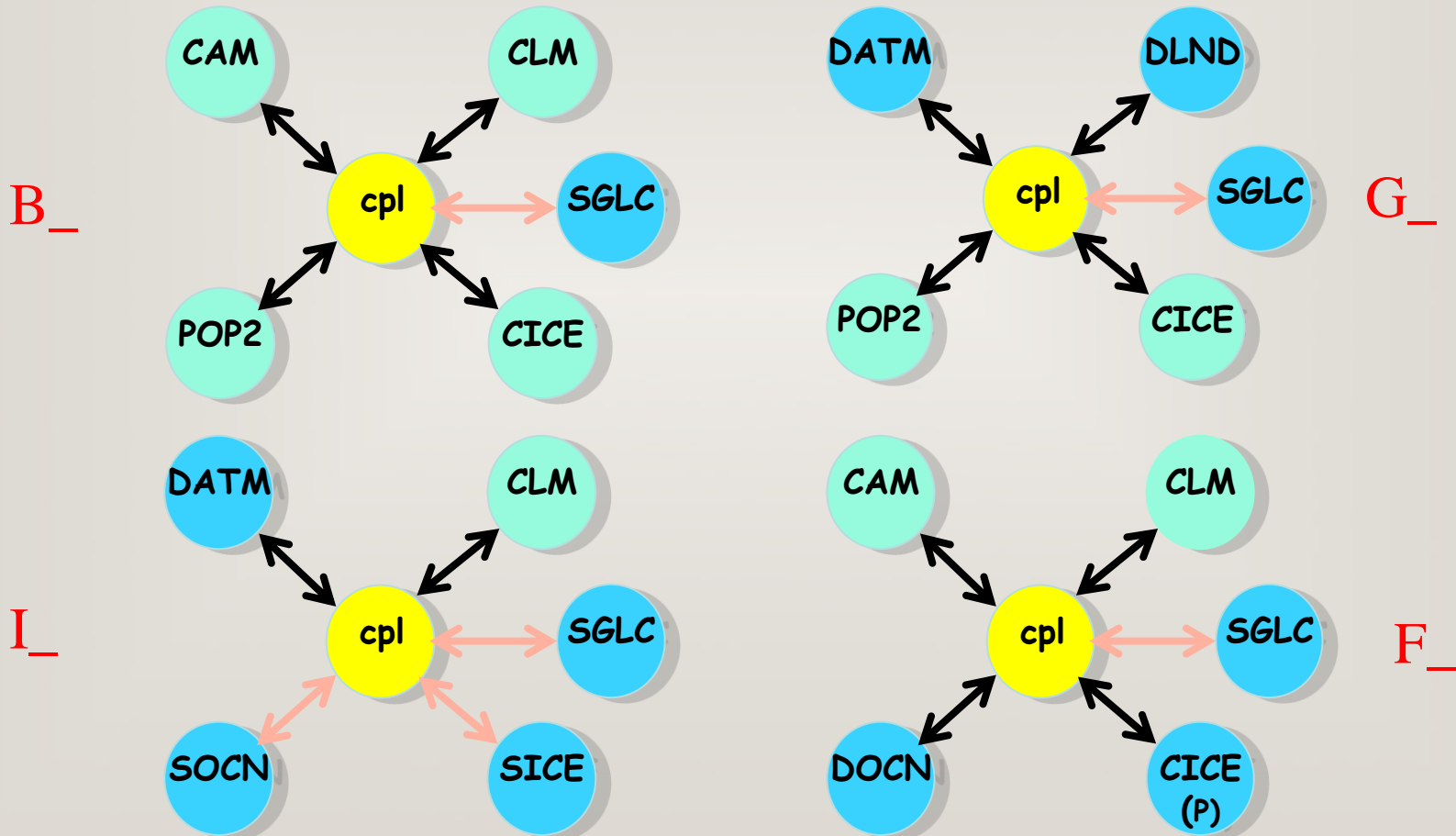
```
A_PRESENT_DAY (A)
  Description: All data model
B_2000 (B)
  Description: All active components, present day
B_1850 (B1850)
  Description: All active components, pre-industrial
F_AMIP (FAMIP)
  Description: Default resolution independent AMIP is INVALID
F_2000_CN (FCN)
  Description: Stand-alone cam default, prescribed ocn/ice with CN
G_NORMAL_YEAR (G)
  Description: Coupled ocean ice with COREv2 normal year forcing
I_2000 (I)
  Description: Active land model with QIAN atm input data for 2003 and Satellite phenology (SP), CO2 level
  and Aerosol deposition for 2000
I_1850 (I1850)
  Description: Active land model with QIAN atm input data for 1948 to 1972 and Satellite phenology (SP), CO2
  level and Aerosol deposition for 1850
```

MACHINES: name (description)

```
bluefire (NCAR IBM p6, os is AIX, 32 pes/node, batch system is LSF)
intrepid (ANL IBM BG/P, os is BGP, 4 pes/node, batch system is cobalt)
jaguarpf (ORNL XT5, os is CNL, 12 pes/node, batch system is PBS)
prototype_ranger (TACC Linux Cluster, Linux (pgi), 1 pes/node, batch system is SGE)
generic_linux_pgi (generic linux (pgi), os is Linux, batch system is PBS, user-defined)
generic_linux_intel (generic linux (intel), os is Linux, batch system is PBS, user-defined)
```

More on CESM component sets

- Plug and play of components (e.g. atm) with different component models (e.g. cam, datm, etc)



(1) Result of Running `create_newcase`

```
./create_newcase -case ~/cases/mycase1 -res f19_g16 -compset B_2000 -mach bluefire
```

```
For both a quick start as well as a detailed summary of creating and running
a CESM model case, see the CESM1.0 User's Guide at
http://www.cesm.ucar.edu/models/cesm1.0

IMPORTANT INFORMATION ABOUT SCIENTIFIC VALIDATION

CESM1.0 has the flexibility to configure cases with many different
combinations of component models, grids, and model settings, but this
version of CESM has only been validated scientifically for the following
fully active configurations:

    1.9x2.5_gx1v6   B_1850_CN
    1.9x2.5_gx1v6   B_1850_RAMPCO2_CN
    1.9x2.5_gx1v6   B_1850-2000_CN

    1.9x2.5_gx1v6   B_1850_CAM5

    .....

please refer to the individual component web pages at
http://www.cesm.ucar.edu/models/cesm1.0

*****
Component set      : B_2000 (B)
Desc               : All active components, present day
*****

Creating ~/cases/mycase1

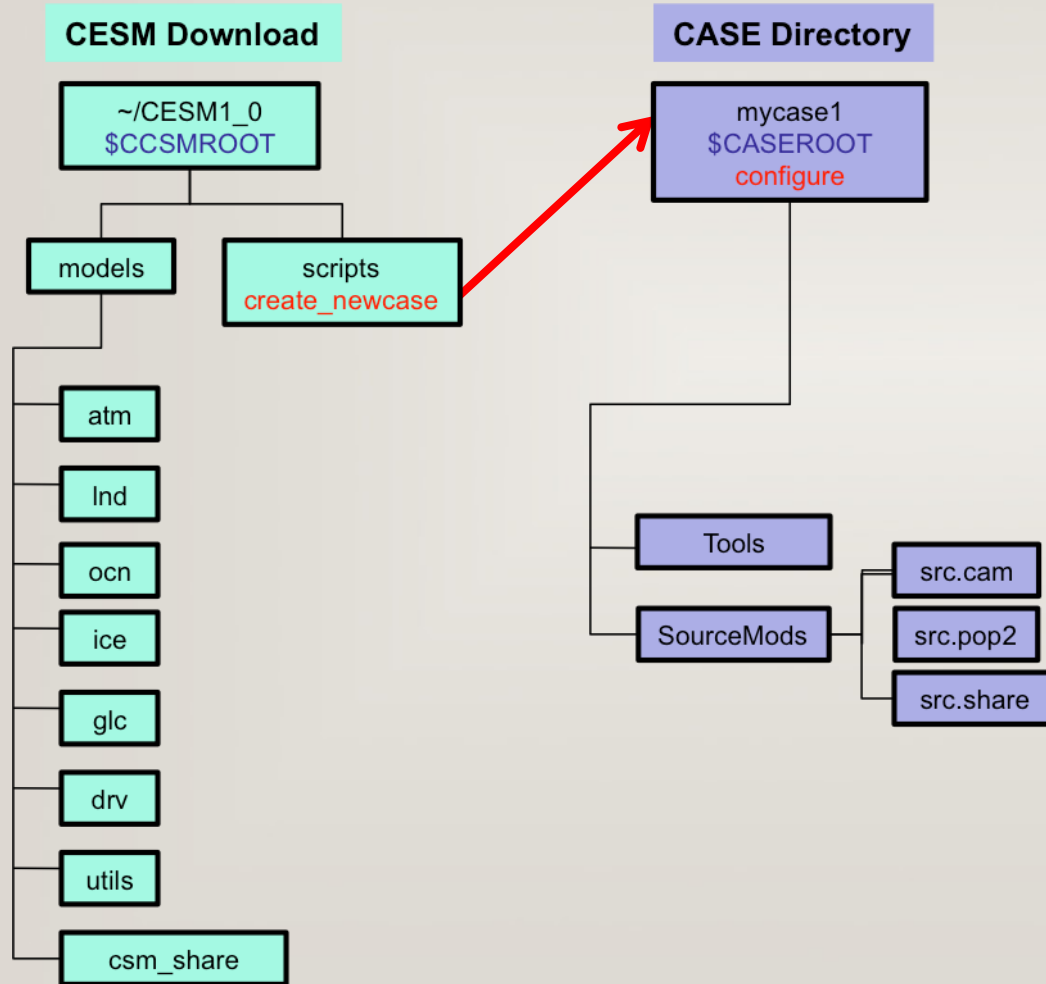
Locking file ~/cases/mycase1/env_case.xml
Successfully created the case for bluefire
```

warning message

case location

success

(1) Overview of Directories (+ create_newcase)



(1) Case Directory After Running `create_newcase`

- `SourceMods` - directory for case specific code modifications
- `configure` - script used in the next step, step (2)
- `env_*.xml` - contains environment variables (more on this later)
- `xmlchange` - script that changes xml (env) variable values

```

CESM1_0/scripts> cd ~/cases/mycase1
cases/mycase1>ls -l
total 64
drwxr-xr-x   2 userx  ncar      8192 May 13 14:32 LockedFiles
-rw-r--r--   1 userx  ncar    10687 May 13 14:32 Macros.bluefire
drwxr-xr-x   2 userx  ncar      8192 May 13 14:32 README
-rw-r--r--   1 userx  ncar       66 May 13 14:32 README.case
drwxr-xr-x   9 userx  ncar      8192 May 13 14:32 SourceMods
drwxr-xr-x   4 userx  ncar      8192 May 13 14:32 Tools
-rwxr-xr-x   1 userx  ncar     9330 May 12 11:33 check_input_data
-rwxr-xr-x   1 userx  ncar    10092 May 12 11:33 configure
-rwxr-xr-x   1 userx  ncar     3085 May 12 11:33 create_production_test
-rw-r--r--   1 userx  ncar     4433 May 13 14:32 env_build.xml
-rw-r--r--   1 userx  ncar     5635 May 13 14:32 env_case.xml
-rw-r--r--   1 userx  ncar     7029 May 13 14:32 env_conf.xml
-rw-r--r--   1 userx  ncar     5915 May 13 14:32 env_mach_pes.xml
-rwxr-xr-x   1 userx  ncar     2199 May 13 14:32 env_mach_specifi
-rw-r--r--   1 userx  ncar    10466 May 13 14:32 env_run.xml
-rwxr-xr-x   1 userx  ncar    10388 May 12 11:33 xmlchange

```

SourceMods

configure

env files

xmlchange

About .xml Files: Format & Variables

- Contains variables used by scripts -- some can be changed by the user
- 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,nst  
eps,nstep,nseconds,nsecond,nminutes,nminute,nhours,nhour,ndays,nday,nmonths,nmonth,nyears,nyea  
r,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" />  
  
<!--"logical to turn on short term archiving, valid values: TRUE,FALSE (logical) " -->  
<entry id="DOUT_S" value="TRUE" />  
  
<!--"local short term archiving root directory (char) " -->  
<entry id="DOUT_S_ROOT" value="/ptmp/$CCSMUSER/archive/$CASE" />
```

- “id” - variable name
 - “value” – variable value
 - <!-- text --> description above the entry
 - To modify a variable in an xml file – use `xmlchange`
 - > `xmlchange -help`
 - > `xmlchange -file env_run.xml -id STOP_N -val 20`
- (Can edit `env_*.xml` file manually -- but be careful about introducing formatting errors)

About .xml Files:

How They Change the Build and Run

- `env_case.xml`
 - Set by `create_newcase` and cannot be modified
- `env_conf.xml`
 - Specifies component information
 - Can change the physics of a model – be very careful about this
- `env_mach_pes.xml`
 - Specifies layout of components on hardware processors
 - Use this to tune performance - **scientific results do not depend on component/processor layout**
- `env_build.xml`
 - Specifies build information
- `Macros.*`
 - Specifies Makefile compilation variables
- `env_mach_specific`
 - Sets modules and paths to libraries (e.g. MPI)
 - Can change compiler options, libraries, etc.
 - Part of porting is to set variables here
- `env_run.xml`
 - Sets run time information
 - User interacts with this file most frequently

```
<entry id="NTASKS_ATM" value="64" />
<entry id="NTHRDS_ATM" value="1" />
<entry id="ROOTPE_ATM" value="0" />

<entry id="NTASKS_LND" value="64" />
<entry id="NTHRDS_LND" value="1" />
<entry id="ROOTPE_LND" value="0" />

<entry id="NTASKS_ICE" value="64" />
<entry id="NTHRDS_ICE" value="1" />
<entry id="ROOTPE_ICE" value="0" />

<entry id="NTASKS_OCN" value="64" />
<entry id="NTHRDS_OCN" value="1" />
<entry id="ROOTPE_OCN" value="0" />

<entry id="NTASKS_CPL" value="64" />
<entry id="NTHRDS_CPL" value="1" />
<entry id="ROOTPE_CPL" value="0" />
```

Basic Work Flow

- **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) **Configure the Case**
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine named "bluefire"

```
# go to root directory of source code download
cd /path/to/source/code/download/cesm1_0/

# go into scripts subdir
cd scripts

# (1) create a new case in your home dir
./create_newcase -case ~/cases/mycase.01 -res f19_g16 -compset B_1850 -
mach bluefire

# go into the case you just created in the last step
cd ~/cases/mycase.01/

# (2) configure the case
./configure -case ←

# (3) build the executable
./mycase.01.bluefire.build

...
```

(2) Configure the Case

- Run `./configure -case`
- Generates
 - Buildconf directory with `buildnml`, `buildexe`, and `input_data_list` files
 - `case *.build` and `*.run` scripts
- Locks `env_conf.xml` and `env_mach_pes.xml`
- Modify `env_conf.xml` and `env_mach_pes.xml` *before running configure*

```

CESM1_0/scripts> cd ~/cases/mycase1
cases/mycase1>ls -l
total 64
drwxr-xr-x    2 userx   ncar           8192 May 13 14:32 LockedFiles
-rw-r--r--    1 userx   ncar          10687 May 13 14:32 Macros.bluefire
drwxr-xr-x    2 userx   ncar           8192 May 13 14:32 README
-rw-r--r--    1 userx   ncar            66 May 13 14:32 README.case
drwxr-xr-x    9 userx   ncar           8192 May 13 14:32 SourceMods
drwxr-xr-x    4 userx   ncar           8192 May 13 14:32 Tools
-rwxr-xr-x    1 userx   ncar          9330 May 12 11:33 check_input_data
-rwxr-xr-x    1 userx   ncar         10092 May 12 11:33 configure
-rwxr-xr-x    1 userx   ncar          3085 May 12 11:33 create_production_test
-rw-r--r--    1 userx   ncar          4433 May 13 14:32 env_build.xml
-rw-r--r--    1 userx   ncar          5635 May 13 14:32 env_case.xml
-rw-r--r--    1 userx   ncar          7029 May 13 14:32 env_conf.xml
-rw-r--r--    1 userx   ncar          5915 May 13 14:32 env_mach_pes.xml
-rwxr-xr-x    1 userx   ncar          2199 May 13 14:32 env_mach_specific
-rw-r--r--    1 userx   ncar         10466 May 13 14:32 env_run.xml
-rwxr-xr-x    1 userx   ncar          10388 May 12 11:33 xmlchange

```

configure

env_conf.xml
env_mach_pes.xml

(2) About configure

> `./configure -help`

NAME

configure - configures the model for a given resolution, component set and machine.

SYNOPSIS

configure [-case] [-cleannamelist] [-cleanmach] [-cleanall]

- `./configure -case`
 - Configures the case
- `./configure -cleanall`
 - Unlocks `env_conf.xml` and `env_mach_pes.xml`
 - “Backs up” `Buildconf/` and run scripts
 - Modify `env_conf.xml` and `env_mach_pes.xml` and type `configure -case` again
- `./configure -cleanmach`
 - Unlocks only `env_mach_pes.xml`
 - “Backs up” run scripts
 - Modify `env_mach_pes.xml` and type `configure -case` again

(2) Running configure

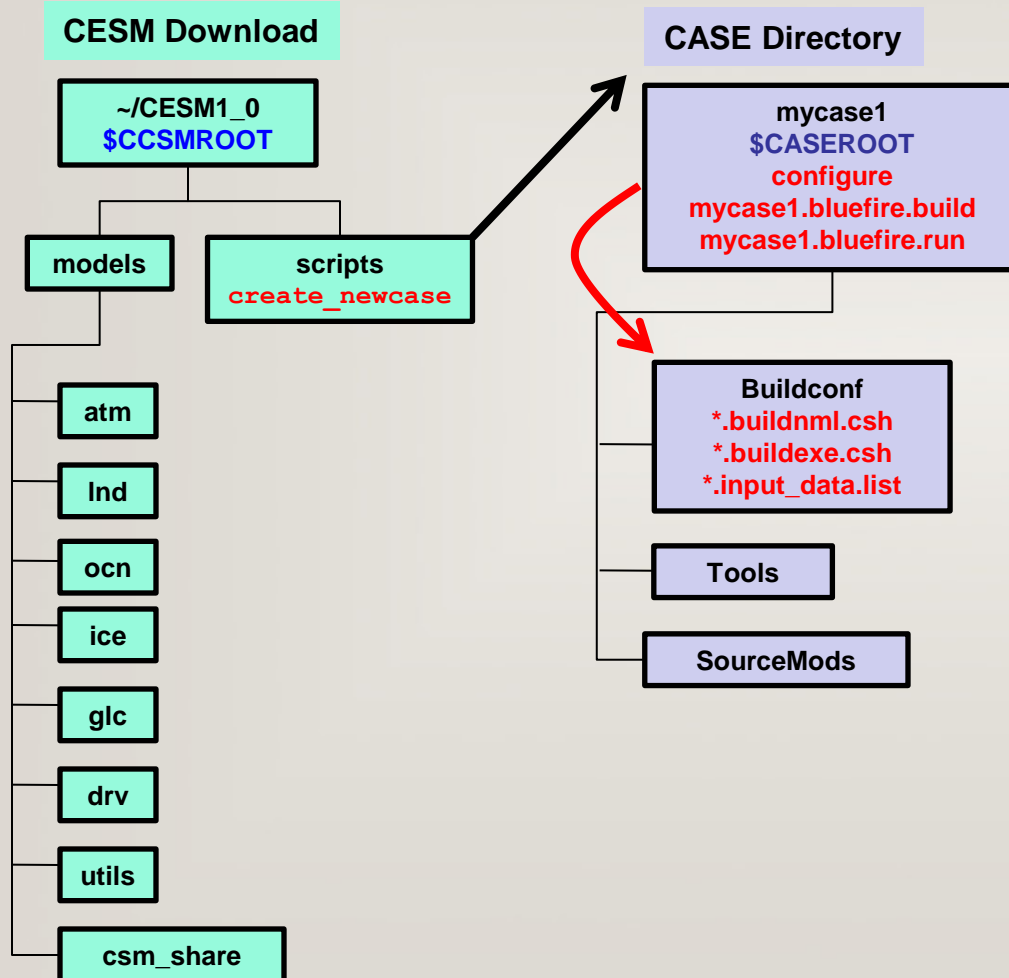
Generated Buildconf files

```
cases/mycase1>./configure -case
Generating resolved namelist, prestage, and build scripts
adding use_case 2000_control defaults for var sim_year with val 2000
adding use_case 2000_control defaults for var sim_year_range with val constant
adding use_case 2000_control defaults for var use_case_desc with val Conditions to simulate 2000
land-use
configure done.
Successfully generated resolved namelist, prestage, and build scripts
Locking file env_conf.xml
Generating clean_build script
Generating build script
Generating run script
Locking file env_mach_pes.xml
Successfully configured the case for bluefire
If an old build exists for this case, you might want to
run the *.clean_build script before building
```

Generated build
and run scripts

Success

(2) Overview of Directories (+ configure)



(2) Case Dir After Running `configure`

- `configure` adds the Buildconf directory and populates it
- `configure` generates build, clean_build, run, and archive scripts

```
cases/mycase1>ls -l
```

```
total 432
drwxr-xr-x   6 userx  ncar      8192 May 13 17:12 Buildconf
drwxr-xr-x   2 userx  ncar      8192 May 13 17:12 LockedFiles
-rw-r--r--   1 userx  ncar    10687 May 13 14:32 Macros.bluefire
drwxr-xr-x   2 userx  ncar      8192 May 13 14:32 README
-rw-r--r--   1 userx  ncar      66 May 13 14:32 README.case
drwxr-xr-x   9 userx  ncar      8192 May 13 14:32 SourceMods
drwxr-xr-x   4 userx  ncar      8192 May 13 14:32 Tools
-rwxr-xr-x   1 userx  ncar     9330 May 12 11:33 check_input_data
-rwxr-xr-x   1 userx  ncar    10092 May 12 11:33 configure
-rwxr-xr-x   1 userx  ncar     3085 May 12 11:33 create_production_test
-rw-r--r--   1 userx  ncar     4454 May 13 17:12 env_build.xml
-rw-r--r--   1 userx  ncar     5635 May 13 14:32 env_case.xml
-rw-r--r--   1 userx  ncar     7029 May 13 14:32 env_conf.xml
-rw-r--r--   1 userx  ncar      614 May 13 17:12 env_derived
-rw-r--r--   1 userx  ncar     5916 May 13 17:12 env_mach_pes.xml
-rwxr-xr-x   1 userx  ncar     2199 May 13 14:32 env_mach_specific
-rw-r--r--   1 userx  ncar    10466 May 13 14:32 env_run.xml
-rwxrwxr-x   1 userx  ncar      574 May 13 17:12 mycase1.bluefire.build
-rwxrwxr-x   1 userx  ncar      836 May 13 17:12 mycase1.bluefire.clean_build
-rwxrwxr-x   1 userx  ncar      802 May 13 17:12 mycase1.bluefire.l_archive
-rwxrwxr-x   1 userx  ncar     3938 May 13 17:12 mycase1.bluefire.run
-rwxr-xr-x   1 userx  ncar    10388 May 12 11:33 xmlchange
```

Buildconf

new scripts

(2) Files in the Buildconf/ Directory (Created by **configure**)

- The configure script fills the Buildconf/ directory which contains
- Component buildnml.csh scripts
- Component buildexe.csh scripts
- Component input_data_list

```
cases/mycase1>ls -l Buildconf/
total 448
-rwxr-xr-x  1 userx  ncar           850 May 13 17:12 cam.buildexe.csh
-rwxr-xr-x  1 userx  ncar        3625 May 13 17:12 cam.buildnml.csh
-rwxr-xr-x  1 userx  ncar        1508 May 13 17:12 cam.input_data_list
drwxr-xr-x  2 userx  ncar        8192 May 13 17:12 camconf
-rwxr-xr-x  1 userx  ncar         480 May 13 17:12 cesm.buildexe.csh
-rwxr-xr-x  1 userx  ncar        1414 May 13 17:12 cice.buildexe.csh
-rwxr-xr-x  1 userx  ncar        3292 May 13 17:12 cice.buildnml.csh
-rwxr-xr-x  1 userx  ncar         379 May 13 17:12 cice.input_data_list
drwxr-xr-x  2 userx  ncar        8192 May 13 17:12 ciceconf
-rwxr-xr-x  1 userx  ncar        1174 May 13 17:12 clm.buildexe.csh
-rwxr-xr-x  1 userx  ncar        2269 May 13 17:12 clm.buildnml.csh
-rwxr-xr-x  1 userx  ncar         702 May 13 17:12 clm.input_data_list
drwxr-xr-x  2 userx  ncar        8192 May 13 17:12 clmconf
-rwxr-xr-x  1 userx  ncar          42 May 13 17:12 cpl.buildexe.csh
-rwxr-xr-x  1 userx  ncar       10507 May 13 17:12 cpl.buildnml.csh
-rwxr-xr-x  1 userx  ncar        1665 May 13 17:12 csm_share.buildlib
-rwxr-xr-x  1 userx  ncar        1965 May 13 17:12 mct.buildlib
-rwxr-xr-x  1 userx  ncar        2412 May 13 17:12 pio.buildlib
-rwxr-xr-x  1 userx  ncar        5546 May 13 17:12 pop2.buildexe.csh
-rwxr-xr-x  1 userx  ncar       29056 May 13 17:12 pop2.buildnml.csh
-rwxr-xr-x  1 userx  ncar        1012 May 13 17:12 pop2.input_data_list
drwxr-xr-x  2 userx  ncar        8192 May 13 17:12 pop2doc
-rwxr-xr-x  1 userx  ncar         588 May 13 17:12 sglc.buildexe.csh
-rwxr-xr-x  1 userx  ncar          78 May 13 17:12 sglc.buildnml.csh
```

Basic Work Flow

- **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) Configure the Case
 - (3) **Build the Executable**
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) Run the Model: Continuation Run(s)

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine named "bluefire"

```
# go to root directory of source code download
cd /path/to/source/code/download/cesm1_0/

# go into scripts subdir
cd scripts

# (1) create a new case in your home dir
./create_newcase -case ~/cases/mycase.01 -res f19_g16 -compset B_1850 -mach bluefire

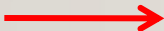
# go into the case you just created in the last step
cd ~/cases/mycase.01/

# (2) configure the case
./configure -case

# (3) build the executable
./mycase.01.bluefire.build

# (4) submit an initial run to the batch queue
bsub < mycase.01.bluefire.run

. . . . .
```



(3) Build the Model

- Use the ***.build** script
- Modifications before build
 - Change `env_build.xml` values *before* running `*.build`
 - Introduce modified source code in `SourceMods/` before building
- To redo build, run ***.clean_build** first
- **The *.build script**
 - Checks for missing input data
 - Creates directory for executable code and model namelist files
 - Locks `env_build.xml`
 - 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

(3) The *.build script

```
cases/mycase1>ls -l
total 432
drwxr-xr-x    6 userx  ncar      8192 May 13 17:12 Buildconf
drwxr-xr-x    2 userx  ncar      8192 May 13 17:12 LockedFiles
-rw-r--r--    1 userx  ncar    10687 May 13 14:32 Macros.bluefire
drwxr-xr-x    2 userx  ncar      8192 May 13 14:32 README
-rw-r--r--    1 userx  ncar      66 May 13 14:32 README.case
drwxr-xr-x    9 userx  ncar      8192 May 13 14:32 SourceMods
drwxr-xr-x    4 userx  ncar      8192 May 13 14:32 Tools
-rwxr-xr-x    1 userx  ncar     9330 May 12 11:33 check_input_data
-rwxr-xr-x    1 userx  ncar    10092 May 12 11:33 configure
-rwxr-xr-x    1 userx  ncar     3085 May 12 11:33 create_production_test
-rw-r--r--    1 userx  ncar     4454 May 13 17:12 env_build.xml
-rw-r--r--    1 userx  ncar     5635 May 13 14:32 env_case.xml
-rw-r--r--    1 userx  ncar     7029 May 13 14:32 env_conf.xml
-rw-r--r--    1 userx  ncar      614 May 13 17:12 env_derived
-rw-r--r--    1 userx  ncar     5916 May 13 17:12 env_mach_pes.xml
-rwxr-xr-x    1 userx  ncar     2199 May 13 14:32 env_mach_specific
-rw-r--r--    1 userx  ncar    10466 May 13 14:32 env_run.xml
-rwxrwxr-x    1 userx  ncar      574 May 13 17:12 mycase1.bluefire.build
-rwxrwxr-x    1 userx  ncar      836 May 13 17:12 mycase1.bluefire.clean_build
-rwxrwxr-x    1 userx  ncar      802 May 13 17:12 mycase1.bluefire.l_archive
-rwxrwxr-x    1 userx  ncar     3938 May 13 17:12 mycase1.bluefire.run
-rwxr-xr-x    1 userx  ncar    10388 May 12 11:33 xmlchange
```

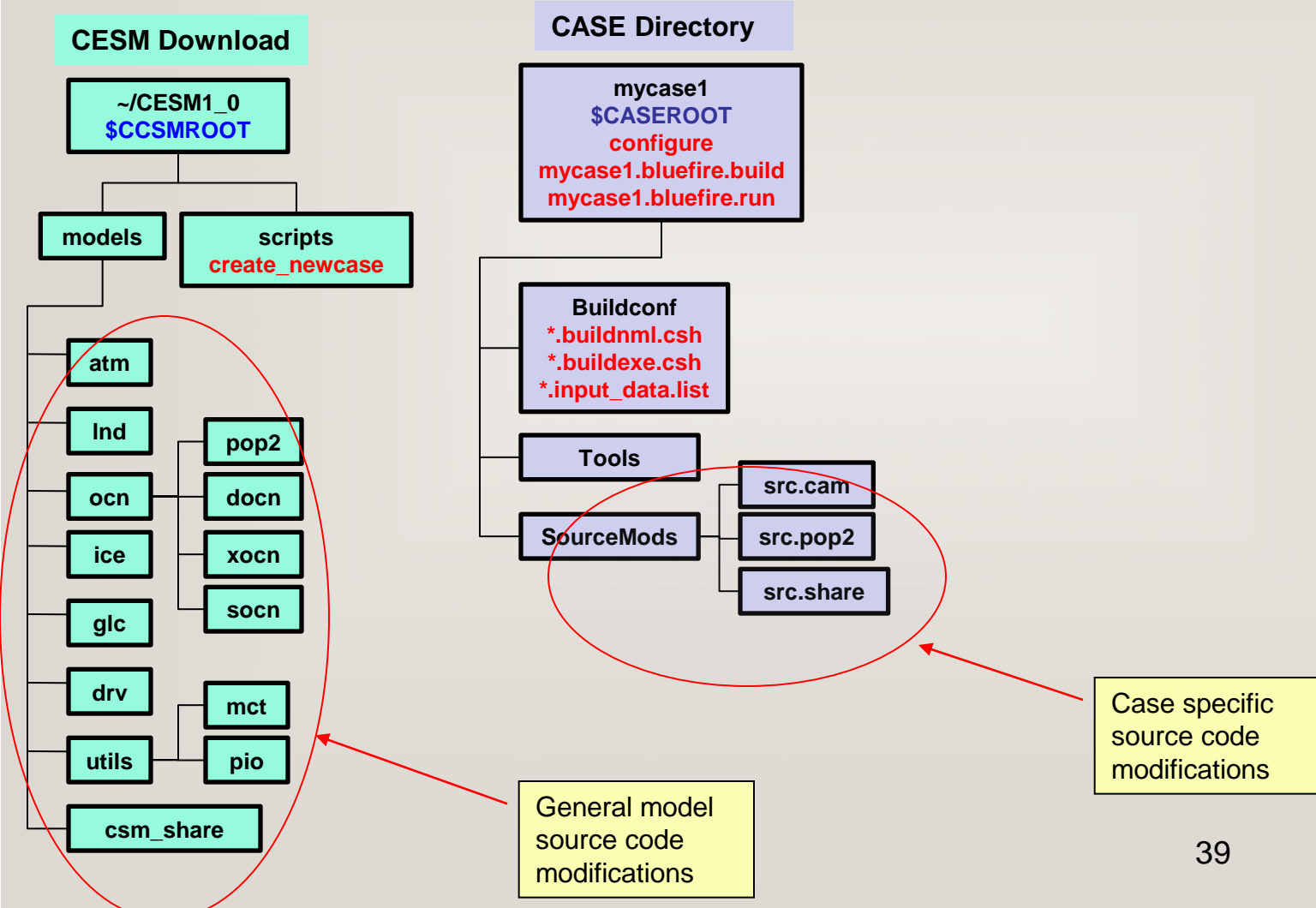
check_input_data

env_build.xml

.build script

(3) Modifying Source Code

Code modified in `models/` will apply to all new cases created – A BAD IDEA
 Modified code in the `CASE SourceMods/` subdirectory applies to that case only
 Files in `SourceMods/` must be in proper subdirectory, eg. `pop2` code in `src.pop2`



(3) 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 ***.buildexe.csh** scripts for each component

```
cases/mycase1>./mycase1.bluefire.build
```

```
-----  
CESM BUILDNML SCRIPT STARTING
```

```
- To prestage restarts, untar a restart.tar file into /ptmp/userx/mycase1/run  
- Create modelio namelist input files
```

```
CESM BUILDNML SCRIPT HAS FINISHED SUCCESSFULLY  
-----
```

```
CESM PRESTAGE SCRIPT STARTING
```

```
- CESM input data directory, DIN_LOC_ROOT_CSMDATA, is /fis/cgd/cseg/csm/inputdata  
- Case input data directory, DIN_LOC_ROOT, is /fis/cgd/cseg/csm/inputdata  
- Checking the existence of input datasets in DIN_LOC_ROOT
```

```
CESM PRESTAGE SCRIPT HAS FINISHED SUCCESSFULLY  
-----
```

```
CESM BUILDEXE SCRIPT STARTING
```

```
- Build Libraries: mct pio csm_share
```

```
Tue May 18 18:06:34 MDT 2010 /ptmp/userx/mycase1/mct/mct.bldlog.100518-180630  
Tue May 18 18:07:21 MDT 2010 /ptmp/userx/mycase1/pio/pio.bldlog.100518-180630  
Tue May 18 18:08:16 MDT 2010 /ptmp/userx/mycase1/csm_share/csm_share.bldlog.10051  
Tue May 18 18:08:59 MDT 2010 /ptmp/userx/mycase1/run/cpl.bldlog.100518-180630  
Tue May 18 18:08:59 MDT 2010 /ptmp/userx/mycase1/run/atm.bldlog.100518-180630  
Tue May 18 18:10:25 MDT 2010 /ptmp/userx/mycase1/run/lnd.bldlog.100518-180630  
Tue May 18 18:11:43 MDT 2010 /ptmp/userx/mycase1/run/ice.bldlog.100518-180630  
Tue May 18 18:12:43 MDT 2010 /ptmp/userx/mycase1/run/ocn.bldlog.100518-180630  
Tue May 18 18:14:52 MDT 2010 /ptmp/userx/mycase1/run/glc.bldlog.100518-180630  
Tue May 18 18:14:53 MDT 2010 /ptmp/userx/mycase1/run/ccsm.bldlog.100518-180630
```

```
- Locking file env_build.xml  
- Locking file Macros.bluefire
```

```
CESM BUILDEXE SCRIPT HAS FINISHED SUCCESSFULLY
```

Namelist
Generation

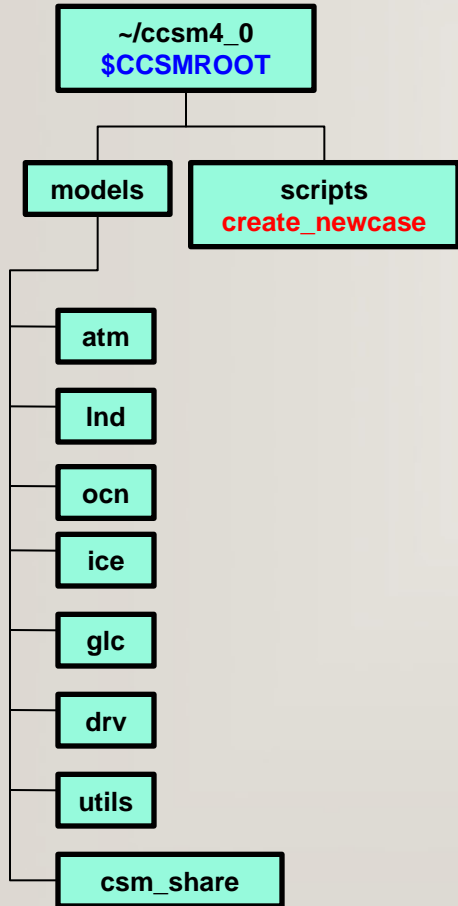
Inputdata
Verification

Model Build

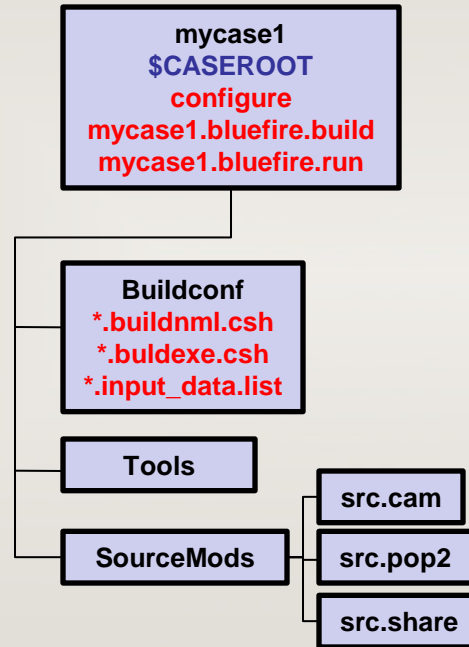
Success

(3) Overview of Directories (+ build)

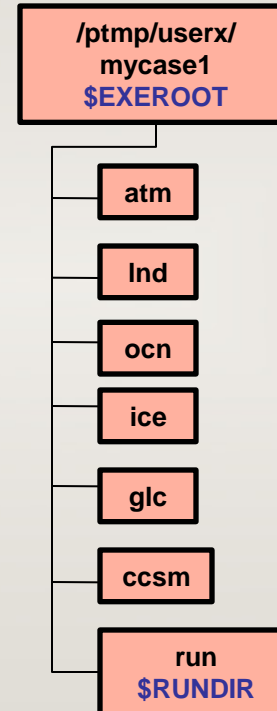
CCSM Download



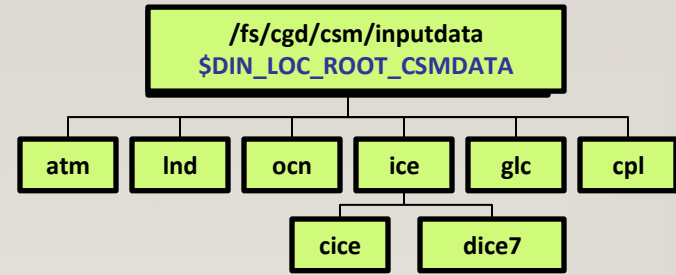
CASE Directory



Build/Run Directory



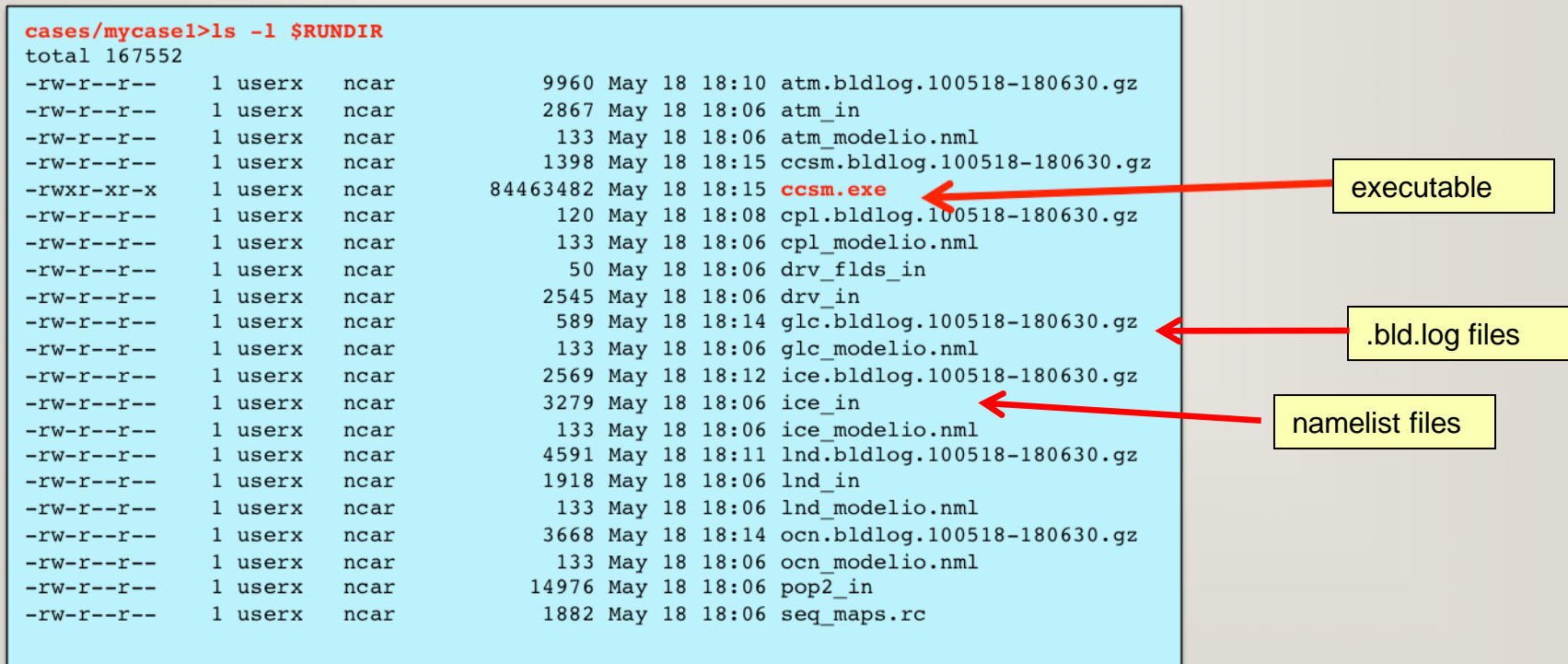
INPUTDATA Directory



(3) Your \$RUNDIR after running .build

```
cases/mycase1> source Tools/ccsm_getenv
cases/mycase1> echo $RUNDIR
> /ptmp/userx/mycase1/run
cases/mycase1> ls -al $RUNDIR
```

```
cases/mycase1>ls -l $RUNDIR
total 167552
-rw-r--r--  1 userx  ncar           9960 May 18 18:10 atm.bldlog.100518-180630.gz
-rw-r--r--  1 userx  ncar           2867 May 18 18:06 atm_in
-rw-r--r--  1 userx  ncar           133 May 18 18:06 atm_modelio.nml
-rw-r--r--  1 userx  ncar          1398 May 18 18:15 ccsm.bldlog.100518-180630.gz
-rwxr-xr-x  1 userx  ncar      84463482 May 18 18:15 ccsm.exe
-rw-r--r--  1 userx  ncar           120 May 18 18:08 cpl.bldlog.100518-180630.gz
-rw-r--r--  1 userx  ncar           133 May 18 18:06 cpl_modelio.nml
-rw-r--r--  1 userx  ncar            50 May 18 18:06 drv_flds_in
-rw-r--r--  1 userx  ncar          2545 May 18 18:06 drv_in
-rw-r--r--  1 userx  ncar           589 May 18 18:14 glc.bldlog.100518-180630.gz
-rw-r--r--  1 userx  ncar           133 May 18 18:06 glc_modelio.nml
-rw-r--r--  1 userx  ncar          2569 May 18 18:12 ice.bldlog.100518-180630.gz
-rw-r--r--  1 userx  ncar          3279 May 18 18:06 ice_in
-rw-r--r--  1 userx  ncar           133 May 18 18:06 ice_modelio.nml
-rw-r--r--  1 userx  ncar          4591 May 18 18:11 lnd.bldlog.100518-180630.gz
-rw-r--r--  1 userx  ncar          1918 May 18 18:06 lnd_in
-rw-r--r--  1 userx  ncar           133 May 18 18:06 lnd_modelio.nml
-rw-r--r--  1 userx  ncar          3668 May 18 18:14 ocn.bldlog.100518-180630.gz
-rw-r--r--  1 userx  ncar           133 May 18 18:06 ocn_modelio.nml
-rw-r--r--  1 userx  ncar          14976 May 18 18:06 pop2_in
-rw-r--r--  1 userx  ncar           1882 May 18 18:06 seq_maps.rc
```



Basic Work Flow

- **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) Configure the Case
 - (3) Build the Executable
 - (4) **Run the Model: Initial Run and Output Data Flow**
 - (5) Run the Model: Continuation Run(s)

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine named "bluefire"

```
# go to root directory of source code download
cd /path/to/source/code/download/cesm1_0/

# go into scripts subdir
cd scripts

# (1) create a new case in your home dir
./create_newcase -case ~/mycase.01 -res f19_g16 -compset B_1850 -mach bluefire

# go into the case you just created in the last step
cd ~/mycase.01/

# (2) configure the case
./configure -case

# (3) build the executable
./mycase.01.bluefire.build

# (4) submit an initial run to the batch queue
bsub < mycase.01.bluefire.run

# check status of job and output files
bjobs
source Tools/ccsm_getenv
ls -lft $RUNDIR
ls -l logs

. . . . .
```

(4) Running the Model: Initial Run

- May want to edit `env_run.xml` file before running (e.g. change run length)
- May also want to modify component namelist settings
- Can change `env_run.xml` variables
- Or directly in the `Buildconf/* .buildnml.csh` files
- The run script
 - Generates the namelist files in `$RUNDIR` (again)
 - Verifies existence of input datasets (again)
 - **DOES NOT build (or re-build) the executable**

```
cases/mycase1> bsub < mycase1.bluefire.run
```

```
cases/mycase1> bsub < mycase1.bluefire.run  
Job <40597> is submitted to queue <regular>.
```

```
cases/mycase1> bjobs
```

```
cases/mycase1> bjobs  
JOBID  USER  STAT  QUEUE      FROM_HOST  EXEC_HOST  JOB_NAME  SUBMIT_TIME  
40597  userx  PEND  regular    bell105en  bell105en  mycase1  May 18 18:30
```

(4) Output in Your CASE Directory

A job completed successfully if "SUCCESSFUL TERMINATION OF CPL7-CCSM" appears near end of the cpl.log file

```

cases/mycase1>ls -l
total 512
drwxr-xr-x  6 userx  ncar          8192 May 18 18:32 Buildconf
drwxr-xr-x  2 userx  ncar          8192 May 18 18:06 CaseDocs
drwxr-xr-x  2 userx  ncar          8192 May 18 18:15 LockedFiles
-rw-r--r--  1 userx  ncar       10687 May 13 14:32 Macros.bluefire
drwxr-xr-x  2 userx  ncar          8192 May 13 14:32 README
-rw-r--r--  1 userx  ncar         66 May 13 14:32 README.case
drwxr-xr-x  9 userx  ncar          8192 May 13 14:32 SourceMods
drwxr-xr-x  4 userx  ncar          8192 May 13 14:32 Tools
-rwxr-xr-x  1 userx  ncar          9330 May 12 11:33 check_input_data
-rwxr-xr-x  1 userx  ncar       10092 May 12 11:33 configure
-rwxr-xr-x  1 userx  ncar       3085 May 12 11:33 create_production_test
-rw-r--r--  1 userx  ncar       4475 May 18 18:32 env_build.xml
-rw-r--r--  1 userx  ncar       5635 May 13 14:32 env_case.xml
-rw-r--r--  1 userx  ncar       7029 May 13 14:32 env_conf.xml
-rw-r--r--  1 userx  ncar        614 May 18 18:37 env_derived
-rw-r--r--  1 userx  ncar       5916 May 13 17:12 env_mach_pes.xml
-rwxr-xr-x  1 userx  ncar       2199 May 13 14:32 env_mach_specific
-rw-r--r--  1 userx  ncar      10466 May 13 14:32 env_run.xml
drwxr-xr-x  3 userx  ncar          8192 May 18 18:37 logs
-rw-r--r--  1 userx  ncar        270 May 18 18:37 poe.stderr.40597
-rw-r--r--  1 userx  ncar       2013 May 18 18:37 poe.stdout.40597
-rwxrwxr-x  1 userx  ncar        574 May 13 17:12 mycase1.bluefire.build
-rwxrwxr-x  1 userx  ncar        836 May 13 17:12 mycase1.bluefire.clean_build
-rwxrwxr-x  1 userx  ncar        802 May 13 17:12 mycase1.bluefire.l_archive
-rwxrwxr-x  1 userx  ncar       3938 May 13 17:12 mycase1.bluefire.run
drwxr-xr-x  2 userx  ncar          8192 May 18 18:37 timing
-rwxr-xr-x  1 userx  ncar      10388 May 12 11:33 xmlchange

cases/mycase1>ls -l logs
total 272
-rw-r--r--  1 userx  ncar      29882 May 18 18:37 atm.log.100518-183212.gz
drwxr-xr-x  2 userx  ncar          8192 May 18 18:15 bld
-rw-r--r--  1 userx  ncar      19115 May 18 18:37 ccsm.log.100518-183212.gz
-rw-r--r--  1 userx  ncar       4998 May 18 18:37 cpl.log.100518-183212.gz
-rw-r--r--  1 userx  ncar      18732 May 18 18:37 ice.log.100518-183212.gz
-rw-r--r--  1 userx  ncar       9384 May 18 18:37 lnd.log.100518-183212.gz
-rw-r--r--  1 userx  ncar      18534 May 18 18:37 ocn.log.100518-183212.gz

cases/mycase1>ls -l timing
total 32
-rw-r--r--  1 userx  ncar       6204 May 18 18:37 ccsm_timing.mycase1.100518-183212
-rw-r--r--  1 userx  ncar      37111 May 18 18:37 ccsm_timing_summary.100518-183212.gz

```

Copies of the Current Namelist Input Files

stdout/err

Log Files

Timing Files

(4) Output in Short Term Archiving Directory

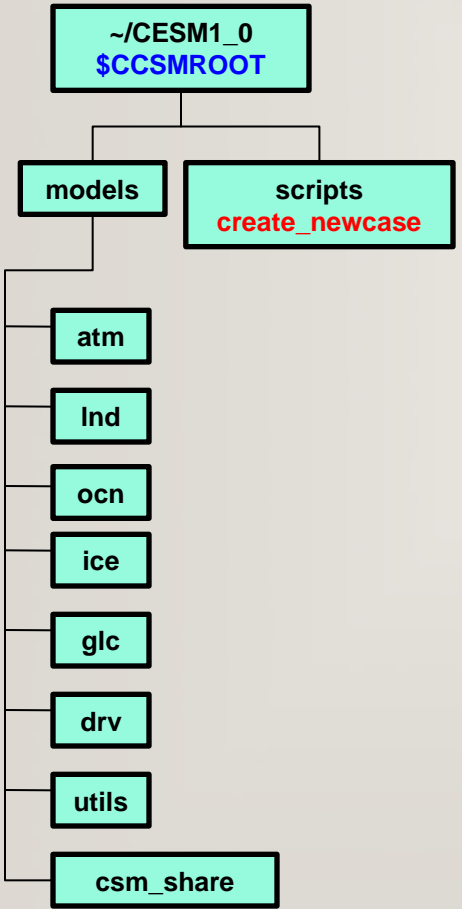
- Output data is originally created in **`$RUNDIR`**
- When the run ends, output data is moved into a short term archiving directory, **`$DOUT_S_ROOT`**
 - Cleans up the `$RUNDIR` directory
 - Migrates output data away from a possibly volatile `$RUNDIR`
 - Gathers data for the long term archive script

```
cases/mycase1>echo $DOUT_S_ROOT
/ptmp/userx/archive/mycase1
cases/mycase1>ls -l $DOUT_S_ROOT
total 1024
drwxr-xr-x   4 userx   ncar           65536 May 18 18:37 atm
drwxr-xr-x   4 userx   ncar           65536 May 18 18:37 cpl
drwxr-xr-x   4 userx   ncar           65536 May 18 18:37 dart
drwxr-xr-x   3 userx   ncar           65536 May 18 18:37 glc
drwxr-xr-x   4 userx   ncar           65536 May 18 18:37 ice
drwxr-xr-x   4 userx   ncar           65536 May 18 18:37 lnd
drwxr-xr-x   4 userx   ncar           65536 May 18 18:37 ocn
drwxr-xr-x   3 userx   ncar           65536 May 18 18:37 rest
cases/mycase1>ls -l $DOUT_S_ROOT/cpl
total 256
drwxr-xr-x   2 userx   ncar           65536 May 18 18:37 hist
drwxr-xr-x   2 userx   ncar           65536 May 18 18:37 logs
cases/mycase1>ls -l $DOUT_S_ROOT/cpl/logs/
total 256
-rw-r--r--   1 userx   ncar           19115 May 18 18:37 ccsml.log.100518-183212.gz
-rw-r--r--   1 userx   ncar           4998  May 18 18:37 cpl.log.100518-183212.gz
cases/mycase1>ls -l $DOUT_S_ROOT/ocn/hist
total 436608
-rw-r--r--   1 userx   ncar              3  May 18 18:32 mycase1.pop.dd.0001-01-02-00000
-rw-r--r--   1 userx   ncar           2787  May 18 18:36 mycase1.pop.do.0001-01-02-00000
-rw-r--r--   1 userx   ncar              3  May 18 18:32 mycase1.pop.dt.0001-01-02-00000
-rw-r--r--   1 userx   ncar           1183  May 18 18:36 mycase1.pop.dv.0001-01-02-00000
-rw-r--r--   1 userx   ncar      27046596 May 18 18:36 mycase1.pop.h.nday1.0001-01-02.nc
-rw-r--r--   1 userx   ncar      78164092 May 18 18:33 mycase1.pop.h.once.nc
-rw-r--r--   1 userx   ncar     117965260 May 18 18:32 mycase1.pop.hv.nc
```

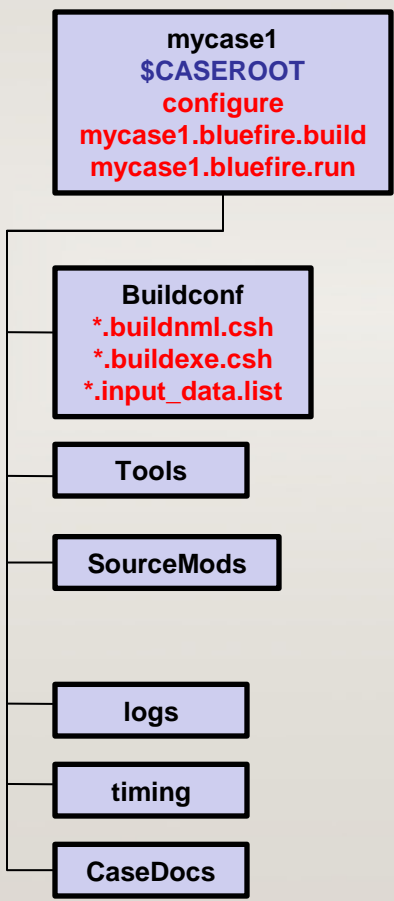
(4) Overview of Directories

(+ run + short term archive)

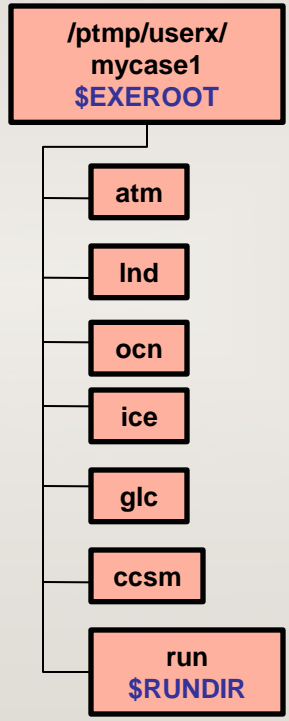
CCSM Download



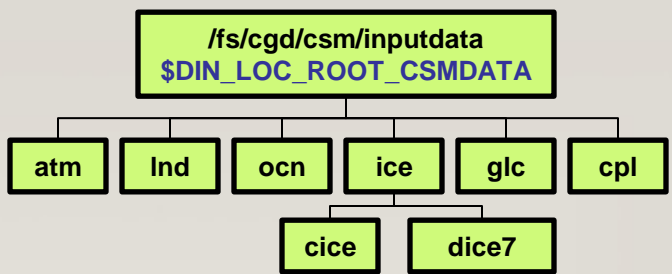
CASE Directory



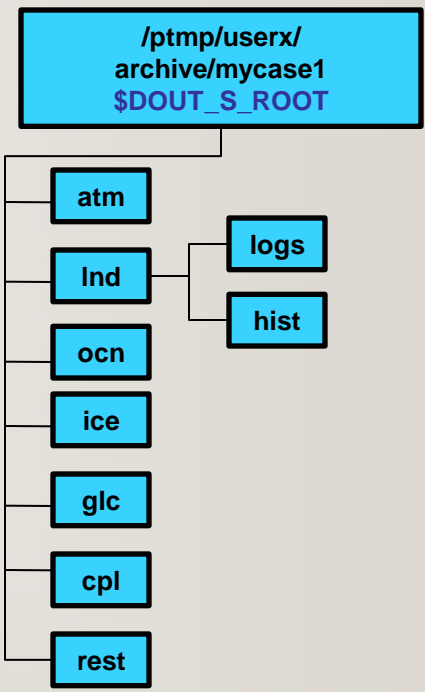
Build/Run Directory



INPUTDATA Directory



Short Term Archive



Basic Work Flow

- **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) Configure the Case
 - (3) Build the Executable
 - (4) Run the Model: Initial Run and Output Data Flow
 - (5) **Run the Model: Continuation Run(s)**

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine named "bluefire"

```
# go to root directory of source code download
cd /path/to/source/code/download/cesm1_0/

# go into scripts subdir
cd scripts

# (1) create a new case in your home dir
./create_newcase -case ~/mycase.01 -res f19_g16 -compset B_1850 -mach bluefire

# go into the case you just created in the last step
cd ~/mycase.01/

# (2) configure the case
./configure -case

# (3) build the executable
./mycase.01.bluefire.build

# (4) submit an initial run to the batch queue
bsub < mycase.01.bluefire.run

# check status of job and output files
bjobs
source Tools/ccsm_getenv
ls -lFt $RUNDIR
ls -l logs

# when the initial run finishes, change to a continuation run
./xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE

# (5) submit a continuation run to the batch queue
bsub < mycase.01.bluefire.run

# check status of job and output files
bjobs
ls -lFt $RUNDIR
ls -l logs
```

(5) Running the Model: Continuation Runs

- Start with a short initial run, described in step (4)
- Examine output to verify that the run is doing what you want
- If the initial run looks good, step (5) is a continuation run

- Change `CONTINUE_RUN` to `TRUE` in `env_run.xml`
- Change `STOP_OPTION` to run the model longer

- May want to turn on auto-resubmit option in `env_run.xml` (`RESUBMIT`)

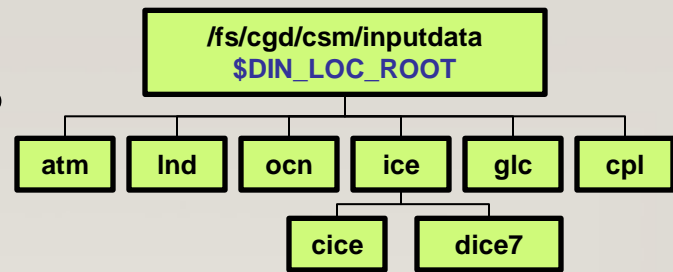
- May want to turn on “long term archiving” in `env_run.xml` (`DOUT_L_MS`)

(5) Long Term Archiving

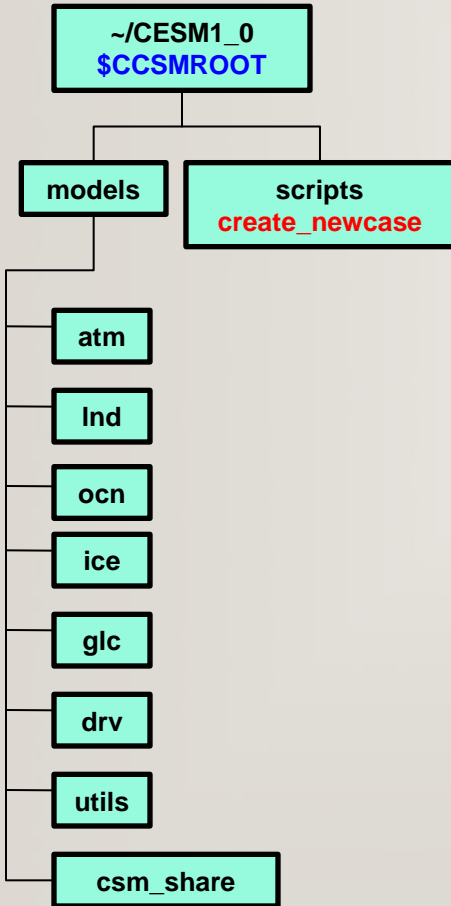
- Why?
 - Migrates output data away from a possibly volatile `$DOUT_S_ROOT` into a permanent long-term storage area
 - Long term archiving script moves data conveniently and in parallel
- To turn on short term archiving (**default is on**)
 - Set `DOUT_S` to `TRUE` in `env_run.xml`
- To turn on long term archiving (**default is off**)
 - Set `DOUT_L_MS` to `TRUE` and set `DOUT_L_MSROOT` in `env_run.xml`
 - Causes run script to automatically submit a long term archiver job (`*.l_archive`) at the end of every successful run.
- Long term archiver
 - Moves data from the short term archive directory to a long term archiving system (e.g. HPSS) - if one exists
 - Runs in batch on one processor
 - Can run in parallel with a production job; will not interfere

(5) Overview of Directories (+ long term archive)

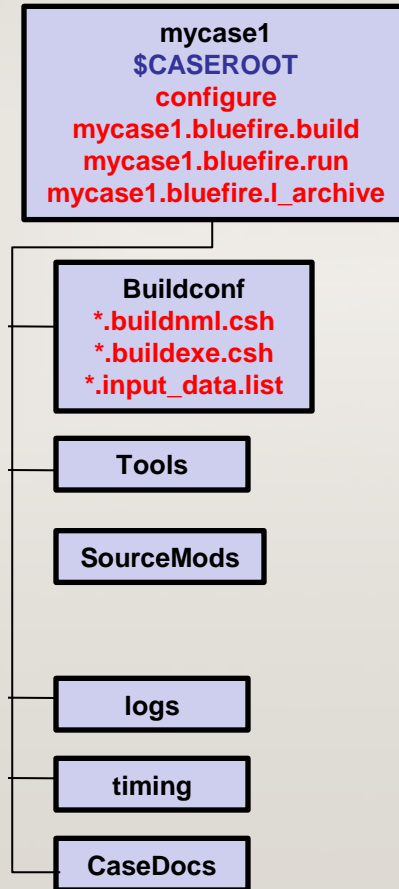
INPUTDATA Directory



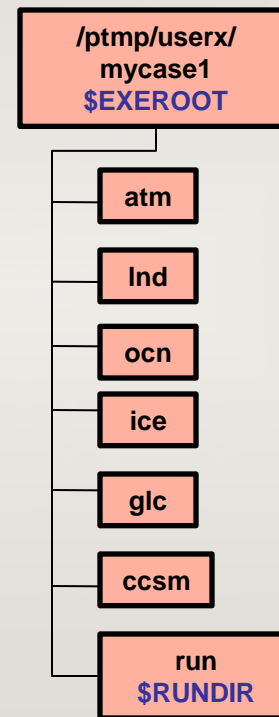
CESM Download



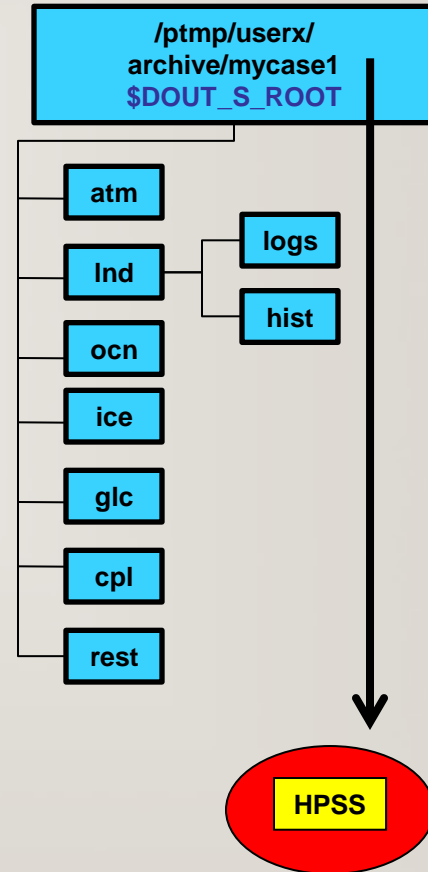
CASE Directory



Build/Run Directory



Short Term Archive



More Information/Getting Help

- **Model User Guides (please provide feedback)**
 - <http://www.cesm.ucar.edu/models/cesm1.0.4/>
 - **CESM Users Guide and Web-Browseable code reference**
 - **CAM, CLM, POP2, CICE, Data Model and CPL7 Users Guides**
- **CESM Bulletin Board**
 - <http://bb.cgd.ucar.edu/>
 - **Facilitate communication among the community**
 - **Ask questions, look for answers**
 - **Many different topics**
- **CESM Release Page Notes**
 - <http://www.cesm.ucar.edu/models/cesm1.0.4/tags/>
 - **Notes significant bugs or issues as they are identified**
- **Model output is available on the Earth System Grid**
 - <http://www.earthsystemgrid.org>
- **Getting Help - email**
 - cesm-help@cgd.ucar.edu
 - **Questions will be answered as resources are available**

Thank You!

Appendix

- A) Steps: Review and Undo
- B) Production Runs
- C) Debugging
- D) Porting
- E) Timing, Performance, Load Balancing
- F) Testing

The NESL 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



Appendix A: Steps, Review and Undo

| Step | “Undo” Command | Associated env Files |
|---------------------------|-----------------------|----------------------------------|
| create_newcase | rm -rf \$CASE & rerun | env_case.xml |
| configure -case | configure -cleanall | env_conf.xml env_mach_pes.xml |
| \$CASE*.build | \$CASE*.clean_build | env_build.xml Macros.* |
| \$CASE*.run | rerun \$CASE*.run | env_run.xml |
| short term archive | set DOUT_S to FALSE | env_run.xml |
| \$CASE*.I_archive | set DOUT_L to FALSE | env_run.xml |

Appendix B: Production Runs

- Verify**

- configuration and inputs
- performance, throughput, cost, and load balance
- exact restart for the production configuration. Use "create_production_test" in the case directory.
- Carry out an initial run and write out a restart set at the end of the run**
- Set STOP_OPTION to "nmonths", set STOP_N
- Set REST_OPTION==STOP_OPTION and REST_N==STOP_N to get a restart at end of run
- When initial run is complete**
- Set CONTINUE_RUN to TRUE in env_run.xml this puts the model in restart mode and the model will start again from the last restart set
- Reset STOP_N to a larger value if appropriate
- Leave REST_OPTION==STOP_OPTION and REST_N==STOP_N
- To turn on short term archiving**
- Set DOUT_S to TRUE in env_run.xml
- To turn on long term archiving**
- Set DOUT_L_MS to TRUE in env_run.xml
- Causes the run script to automatically submit a long term archiver job at the end of every successful run. The long term archiver moves data from the short term archive directory to a mass storage system, runs in batch on one processor, can run in parallel with a production job, and will not interfere with a production job or vice versa.
- To turn on the auto resubmit feature**
- Set RESUBMIT to an integer > 0 in env_run.xml; this causes the run script to resubmit itself after a successful run and decrement the RESUBMIT variable by 1. The model will automatically resubmit until the RESUBMIT variable is decremented to 0.

Appendix C: Debugging

- The CESM scripts will trap invalid env variable values and types when possible and produce an error message
- The scripts can detect when the model needs to be re-configured or re-built due to changes in setup (env and Macros) files and an error message will be produced.
- If input data is not available locally, it will be downloaded automatically. If that data is not available on the CESM input data server, an error message will be produced.
- “**configure –cleanall**” backs up the build, run, and Buildconf files and resets them to original values. Manual changes to namelist values or batch submission settings will be lost in this step and have to be reimplemented manually. The old copies are placed under the MachinesHist directory in the case directory.
- If the build step fails, an error message will be produced and point users to a specific build log file.
- If a run does NOT complete properly, the stdout file often produces an error message like “Model did not complete – see ../cpl.log...”. That cpl log file is associated with the run but may not contain a relevant error message. All the log files will need to be reviewed.
- If a run does NOT complete properly, short term archiving is NOT executed and the timing files are NOT generated. In addition, log files are NOT copied into the case logs directory. Review the stdout/stderr files in the case directory and “cd” to the \$RUNDIR directory and systematically check the latest log files for error messages.
- If a run does NOT complete properly, check whether it timed out because it hit the batch time limit. If it hit the time limit, does it appear to have been running when it timed out or did it hang before it timed out? Check the timestamps on the log files in \$RUNDIR and check the timestamps of the daily timers in the cpl.log file.

Appendix D: Porting – Machines Directory

- Go to the scripts directory
- ccsm_utils/Machines contains machine specific information, porting changes will occur there

```
CESM1_0/scripts>ls -l
total 400
-rw-r--r--  1 userx  ncar          18596 May 12 11:33 ChangeLog
-rw-r--r--  1 userx  ncar           168 May 12 11:33 README
-rw-r--r--  1 userx  ncar           103 May 12 11:33 SVN_EXTERNAL_DIRECTORIES
drwxr-xr-x 10 userx  ncar          8192 May 12 11:33 ccsm_utils
-rwxr-xr-x  1 userx  ncar         19039 May 12 11:33 create_clone
-rwxr-xr-x  1 userx  ncar        52338 May 12 11:33 create_newcase
-rwxr-xr-x  1 userx  ncar        18253 May 12 11:33 create_test
-rwxr-xr-x  1 userx  ncar         9643 May 12 11:33 create_test_suite
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 doc
-rwxr-xr-x  1 userx  ncar         1255 May 12 11:33 link_dirtree
-rw-r--r--  1 userx  ncar           295 May 12 11:33 sample_compset_file.xml
-rw-r--r--  1 userx  ncar           851 May 12 11:33 sample_pes_file.xml
```

ccsm_utils

```
CESM1/scripts>ls -l ccsm_utils
total 112
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 Build
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 Case.template
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 Components
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 Machines
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 Testcases
drwxr-xr-x  3 userx  ncar          8192 May 12 11:33 Testlists
drwxr-xr-x  5 userx  ncar          8192 May 12 11:33 Tools
```

Machines

Appendix D (cont): Porting - Methods

- This is not as easy as we'd like it to be; see the CESM1 Users Guide for more details
- Generic Machine Method**
- Create a case using one of the generic machines and the following create_newcase options
 - scratchroot (generic high level run directory, ie. /ptmp/userx)
 - din_loc_root_csmdata (location of inputdata directory)
 - max_tasks_per_node (max number of mpi tasks per node)
- configure -case**
- Search for "GENERIC_USER", read those comments and edit as needed
- Macros.* (for build settings)
- env_mach_specific (for local machine settings)
- *.run (for batch and launch settings)
- Once things are working, back port the mods to some new machine specific scripts (see below)
- Specific Machine Method**
- cd scripts/ccsm_utils/Machines
- copy a set of "close" machine specific files to your machine name
 - cp Macros.bluefire Macros.mine
 - cp env_machopts.bluefire env_machopts.mine
 - cp mkbatch.bluefire mkbatch.mine
- Edit those files based on your machine configuration as best as you can
- Add an entry in config_machines.xml for your machine (copy and paste) and edit those variables based on your machine
- cd back to the script directory
- create a case using your new machine entry (ie. mine) and test (see above)

Appendix D (cont): Porting - Tips

- Review your local-machine specific documentation and be prepared to reference it as you proceed.
- Prior to starting, know which compiler you want to use and review batch submission and job launching, especially for MPI jobs
- netcdf needs to be installed and you need to know how to link to it.
- With both generic or machine specific approaches, the goal is the same. You want to be able to setup and run CESM cases "out of the box" on your local machine and to be able to share that capability between multiple people. In other words, you want to port to a new machine once and be (mostly) done.
- With either the generic or machine specific approach, there will likely be several iterations between testing the case and updating the machine specific files before things work "out of the box". Be patient.
- Start with an X compset and demonstrate an ability to create a case, build, and run "out of the box" before moving on to more complex configurations. X is an "all dead" configuration that is fast, requires minimal memory, requires minimal input datasets, runs in relatively arbitrary processor layouts, and will test the full coupler implementation with MPI.
- Generally, CESM builds all the components using a single CESM specified Makefile and Macros file. The exceptions are MCT and PIO which are built on the fly in CESM but leverage their own build systems. If you are having problems building MCT or PIO, look for the string "CONFIG_ARGS" in the Macros file. There are independent CONFIG_ARGS for MCT and PIO. Those config_arg options are passed to the MCT and PIO build systems. See other machines Macros files in `scripts/ccsm_utils/Machines` for examples of the CONFIG_ARGS settings on other machines and review the MCT or PIO build logs for errors. Errors generally occur more often in the configure/setup step of the MCT and PIO build than in the build itself.
- Reference the CESM users guide and see current implementations for other machines in the [scripts/ccsm_utils/Machines](#) directory.

Appendix E: Timing

- **env_mach_pes.xml** sets the component pe layout, to change it
- **modify env_mach_pes.xml**

```
mycase1> ./configure --cleanmach
mycase1> ./configure --case
mycase1> ./mycase1.bluefire.clean_build
mycase1> ./mycase1.bluefire.build
mycase1> bsub < mycase1.bluefire.run
```

- **Timing Files**

- See **mycase1/logs/cpl.log*** file to verify completion and get throughput, basic timing and memory output. **cpl.log*** also provides timing for each model day run so temporal variability in cost can be assessed.
- See **mycase1/timing/ccsm_timing.mycase1.*** file for throughput and load balance (next slide)
- See **mycase1/timing/ccsm_timing_summary.*** for individual rawer model timing output

```
mycase1>tail -20 logs/cpl.log.100519-210440
```

```
tStamp_write: model date = 10120      0 wall clock = 2010-05-19 21:11:07 avg dt = 16.43 dt = 16.12
tStamp_write: model date = 10121      0 wall clock = 2010-05-19 21:11:23 avg dt = 16.43 dt = 16.34

(seq_mct_drv): ===== SUCCESSFUL TERMINATION OF CPL7-CCSM =====
(seq_mct_drv): ===== at YMD,TOD = 10121 0 =====
(seq_mct_drv): ===== # simulated days (this run) = 20.000 =====
(seq_mct_drv): ===== compute time (hrs) = 0.091 =====
(seq_mct_drv): ===== # simulated years / cmp-day = 14.410 =====
(seq_mct_drv): ===== pes min memory highwater (MB) 324.382 =====
(seq_mct_drv): ===== pes max memory highwater (MB) 787.038 =====
```

Appendix E (cont): Performance & Load Balance

- Load Balance

- Set STOP_OPTION to 'ndays', STOP_N to 20, REST_OPTION to 'never'

mycase1>cat timing/ccsm_timing.mycase1.100519-210440

| component | comp_pes | root_pe | tasks | x threads |
|------------|----------|---------|-------|-----------|
| cpl = cpl | 48 | 0 | 48 | x 1 |
| glc = sglc | 64 | 0 | 64 | x 1 |
| lnd = clm | 64 | 0 | 16 | x 4 |
| ice = cice | 128 | 16 | 32 | x 4 |
| atm = cam | 192 | 0 | 48 | x 4 |
| ocn = pop2 | 64 | 48 | 64 | x 1 |

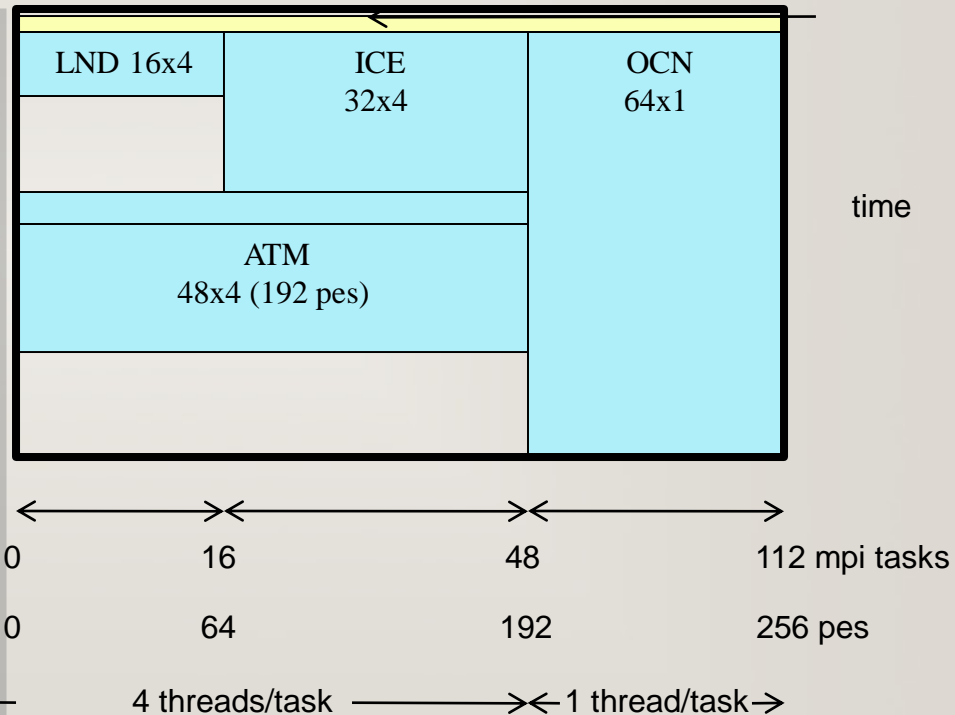
total pes : 256

Overall Metrics:

Model Cost: 213.62 pe-hrs/simulated_year
 Model Throughput: 14.38 simulated_years/day

Init Time : 56.365 seconds
 Run Time : 329.209 seconds 16.460 seconds/day
 Final Time : 0.071 seconds

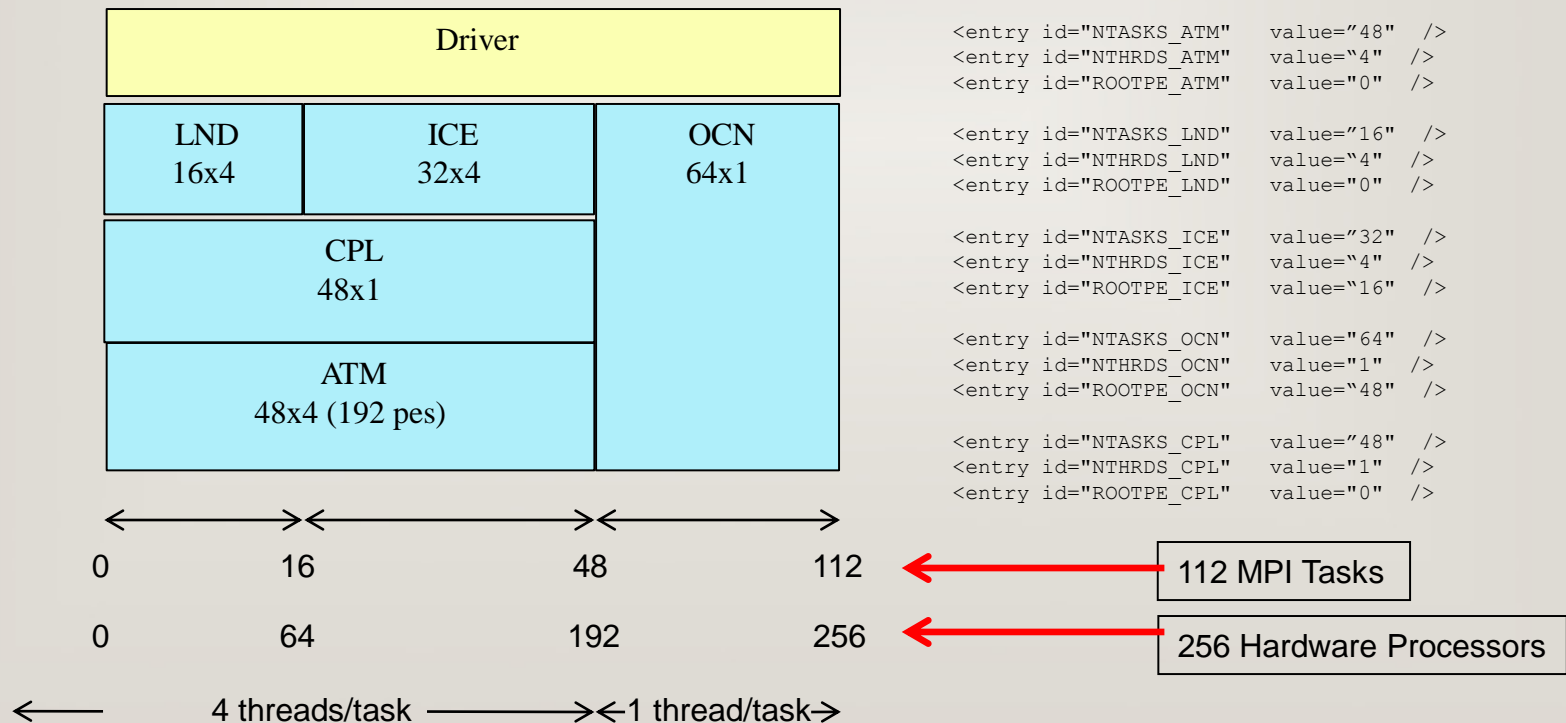
TOT Run Time: 329.209 seconds 16.460 seconds/mday
 LND Run Time: 23.406 seconds 1.170 seconds/mday
 ICE Run Time: 122.689 seconds 6.134 seconds/mday
 ATM Run Time: 107.499 seconds 5.375 seconds/mday
 OCN Run Time: 328.911 seconds 16.446 seconds/mday
 GLC Run Time: 0.000 seconds 0.000 seconds/mday
 CPL Run Time: 13.536 seconds 0.677 seconds/mday



Appendix E (cont): Load Balancing & env_mach_pes.xml

- Some env_mach_pes.xml variables are
- NTASKS_* - number of mpi tasks assigned to the component
- NTHRDS_* - number of openmp threads per mpi task for the component
- ROOTPE_* - global mpi task rank of the component root mpi task

FOR EXAMPLE:



Appendix F: Testing

- **create_production_test** script in the case directory
- **create_test** in the scripts directory
- Generates an automated CESM test
- **create_test -list** produces a list of available tests
- To use:

```
CESM1_0/scripts> ./create_test -testname ERS.f19_g16.X.bluefire -testid t1  
CESM1_0/scripts> cd ERS.f19_g16.X.bluefire.t1  
CESM1_0/scripts/ERS.f19_g16.X.bluefire.t1> ./ERS.f19_g16.X.bluefire.t1.build  
CESM1_0/scripts/ERS.f19_g16.X.bluefire.t1> bsub < ERS.f19_g16.X.bluefire.t1.test  
CESM1_0/scripts/ERS.f19_g16.X.bluefire.t1> cat TestStatus
```

```
CESM1_0/scripts/ERS.f19_g16.X.bluefire.t1>cat TestStatus  
PASS ERS.f19_g16.X.bluefire
```

- **create_test_suite** in the scripts directory
- to generate a suite of tests listed in a specific file

Note: DO NOT submit
the run script, submit
the test script