



CESM1 Tutorial

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

CCSM 4.0 (released April 2010)

CESM 1.0 (released June 2010)

.....

CESM1.0.4 (released Feb 2012)

NCAR is sponsored by the National Science Foundation



Outline

- **Release Homepage on Web**
- **Software & Hardware Requirements**
- **Basic Work Flow**
- **One-Time Setup**
 - **Registration and Source Code Download**
 - **Create an Input Data Root Directory**
 - **Porting**
- **Creating & Running a Case**
 - **Create a New Case**
 - **Configure the Case**
 - **Build the Executable**
 - **Initial Run and Output Data**
 - **Continuation Runs**
- **Getting More Help**
- **Appendix**



CESM1 Release Web Page

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

Notable Improvements



Data, Diagnostics, and Post-Processing Tools



User's Guide



Component Model Documentation



External Libraries



Input Data



Timing Table



The screenshot shows the CESM1 Release Web Page with a navigation bar at the top containing links for Administration, Working Groups, Models, Events, News, and Publications. Below the navigation bar is a search bar and a 'Community Earth System Model' header. The main content area is divided into several sections:

- ABOUT CESM 1.0**: A text block describing the CESM1.0 model.
- MODEL OUTPUT DATA AND DIAGNOSTICS**: A list of links including Model Output Diagnostic Plots, Model Output Data (ESG), and Post-Processing Utilities.
- MODEL DOCUMENTATION**: A grid of links for various model components:
 - CESM1.0 User's Guide**
 - Atmosphere Models**: Community Atmosphere Model (CAM), Climatological Data Model (CDM), and Community Ice Code (CICE).
 - Land Models**: Community Land Model (CLM) and Climatological Data Model (CDM).
 - Sea Ice Models**: Community Ice Code (CICE) and Climatological Ice Model (CIM).
 - Ocean Models**: CESM POP (POP), Climatological Slab-Ocean Data Model (COSM), and Community Ice Sheet Model (CISM).
 - Land Ice Models**: Community Ice Sheet Model (CISM).
 - CESM Coupler**: CESM Coupler (CPL).
- External Library Documentation**: A list of links for Parallel I/O Library (PIO), Model Coupling Toolkit (MCT), and Earth System Modeling Framework (ESMF).
- MODEL INPUT DATA**: A text block explaining the input data repository and how to access it.
- PERFORMANCE AND LOAD BALANCING DATA**: A text block explaining the CESM1 Timing Table.

On the right side of the page, there are several sections:

- CESM PROJECT**: A text block describing the CESM project and its sponsors.
- MODEL SOURCE CODE**: A section containing 'Copyright and Terms of Use' and 'Acquiring the Code'.
- Version Summaries and Known Problems**: A section containing a table of available versions of code and their test record.
- Reporting a Problem**: A section containing instructions on how to report a problem.

Background and Sponsors



Copyright and Terms of Use



How to Acquire the Code



Version Summaries



Known Problems and Reporting 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, openmpi or mpich seem to work on linux clusters)**

- **CESM currently runs on “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**
 -
• **and a few others**



Basic Work Flow

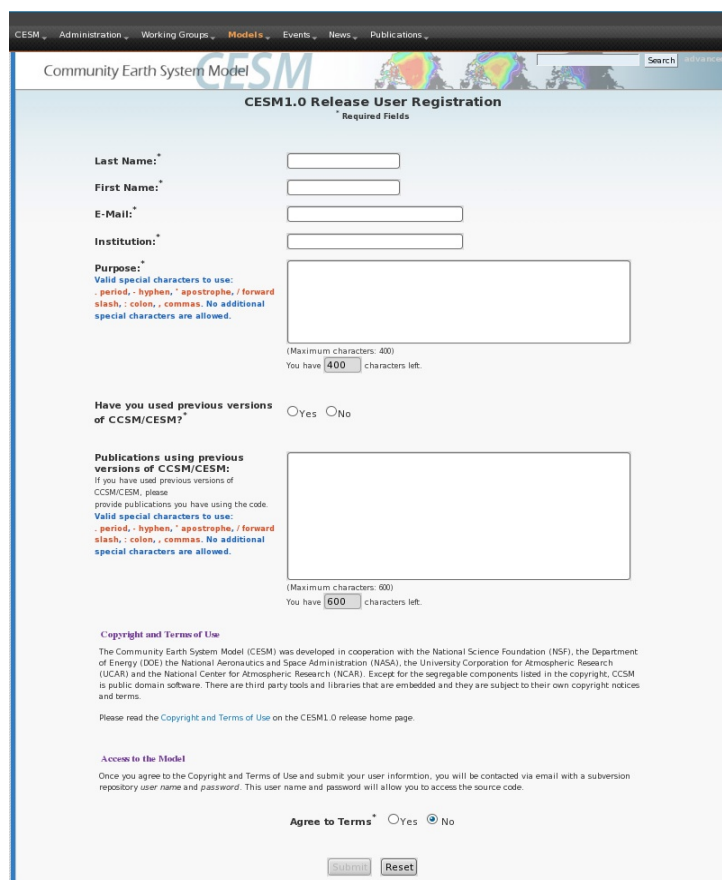
or how to set up and run an experiment

- **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 bar with links for 'Administration', 'Working Groups', 'Models', 'Events', 'News', and 'Publications'. Below this is the 'Community Earth System Model' header with the CESM logo and a search bar. The main content area is titled 'CESM1.0 Release User Registration' and includes a 'Required Fields' section with the following form elements:

- Last Name:** Text input field.
- First Name:** Text input field.
- E-Mail:** Text input field.
- Institution:** Text input field.
- Purpose:** Text area with a character count of 400 (400 characters left).
- Have you used previous versions of CCSM/CESM?:** Radio buttons for 'Yes' and 'No'.
- Publications using previous versions of CCSM/CESM:** Text area with a character count of 600 (600 characters left).

Below the form, there is a 'Copyright and Terms of Use' section with a link to read the full terms. At the bottom, there is an 'Agree to Terms' section with radio buttons for 'Yes' and 'No' (selected), 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

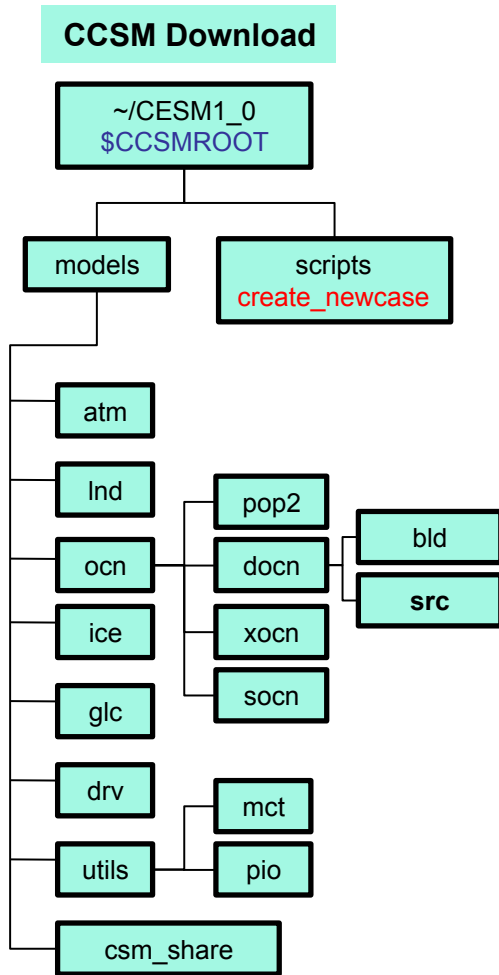
password

> **svn co --username guestuser**

https://svn-ccsm-release.cgd.ucar.edu/models_versions/cesm1_0_4



(A) Overview of Directories (initial model download)





Next Step In the 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***

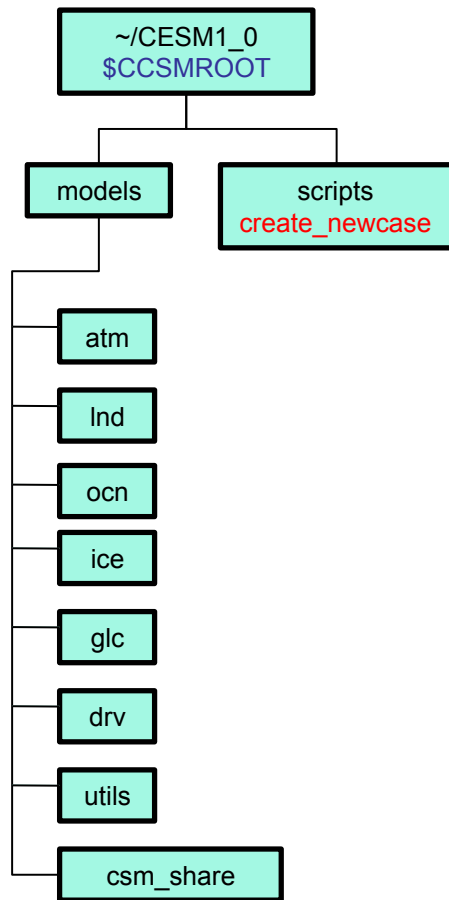


(B) Overview of Directories (+ inputdata directory)

INPUTDATA Directory

/fs/cgd/csm/inputdata
\$DIN_LOC_ROOT_CSMDATA

CESM Download





Next Step In the 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/cesml_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
```



Next Step In the 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 Runs



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/

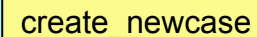
# (2) configure the case
configure -case

.....
```


(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 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
```



create_newcase



(1) About `create_newcase`

- `create_newcase` has many command line options - most are rarely used
- `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.

```
-case <name>           Specifies the case name (required).
-compset <name>       Specify a CESM compset (required).
-res <name>           Specify a CCSM grid resolution (required).
-mach <name>          Specify a CESM machine (required).
-pecount <name>        Value of S,M,L,X1,X2 (optional). (default is M).
-pes_file <name>       Full pathname of pes setup file to use (will overwrite default settin
-compset_file <name>   Full pathname of compset setup file to use. (optional)

-help [or -h]          Print usage to STDOUT (optional).
-list                  Only list valid values for compset, grid settings and machines (optional).
-silent [or -s]        Turns on silent mode - only fatal messages issued (optional).
-verbose [or -v]       Turn on verbose echoing of settings made by create_newcase (optional).
-xmlmode <name>        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.

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

The following two arguments turn on single point mode.

If one is given -- both MUST be given.

```
-pts_lat <value>        Latitude of single point to operate on (optional)
-pts_lon <value>        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], eg. , f19_g16 is 1.9x2.5 atm/lnd + gx1v6 ocn/ice
 - Most often the atm & lnd share the same grid, and the ice & ocn share the same 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
 - Eg. “B” compsets use all active models (CAM,CLM,CICE,POP2)
 - Eg. “F” compsets use CAM,CLM, CICE(prescribed-thermo), DOCN(prescribed-SST)
 - 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
- **create_newcase -list**
 - lists all the valid choices for these command line options
- **values are set on the command line are “locked down” in case directory**
 - file env_case.xml contains all “locked down” variables when create_newcase was run



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

```
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)
4x5_gx3v7 (f45_g37)
T31_gx3v7 (T31_g37)
ne30np4_1.9x2.5_gx1v6 (ne30_f19_g16)
```

COMPSETS: name (shortname): description (status)

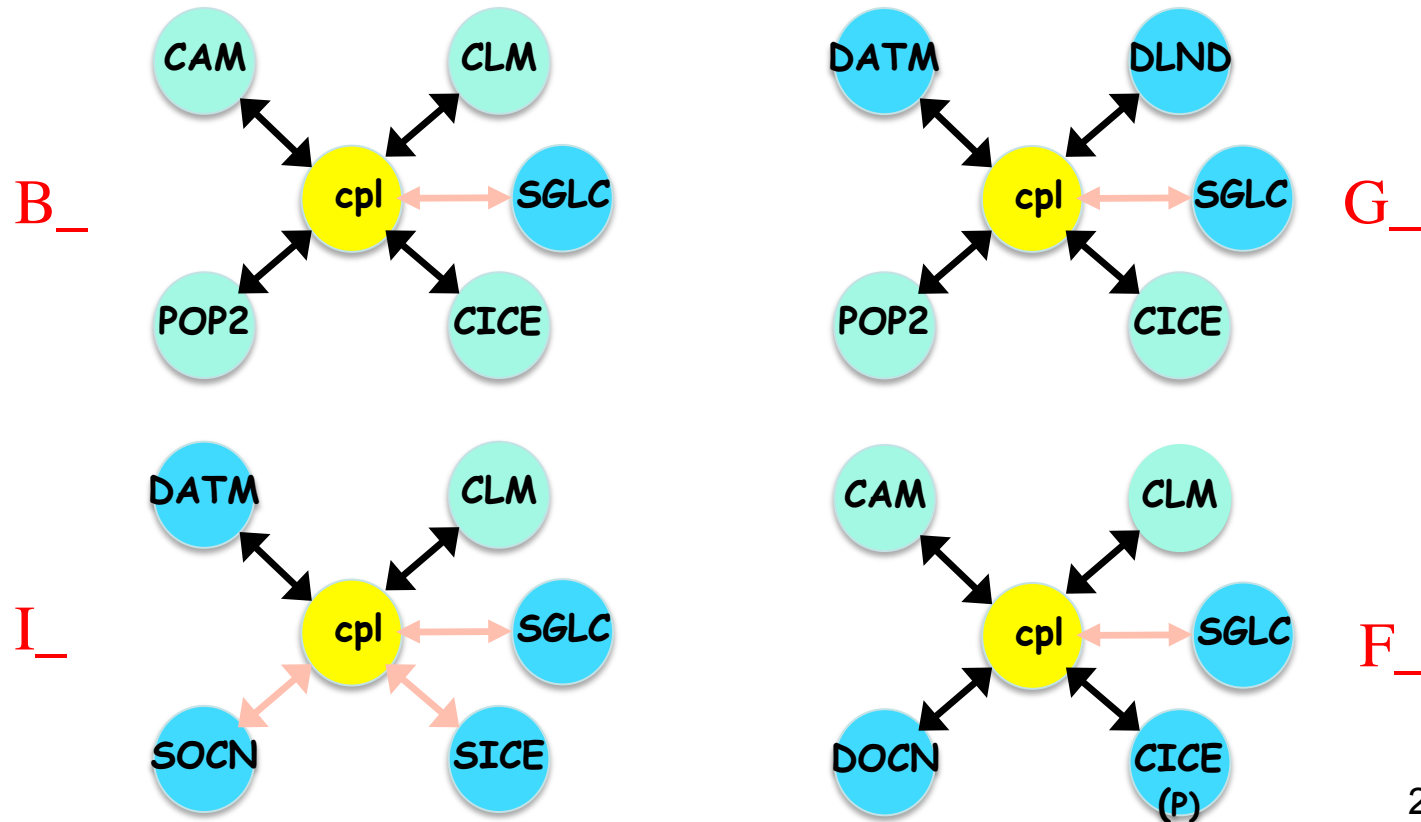
```
A_PRESENT_DAY (A)
  Description: All data model
B_2000 (B)
  Description: All active components, present day
B_1850 (B1850)
  Description: All active components, pre-industrial
B_1850_CN (B1850CN)
  Description: all active components, pre-industrial, with CN (Carbon Nitrogen) in CLM
F_AMIP (FAMIP)
  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)
franklin (NERSC XT4, os is CNL, 4 pes/node, batch system is PBS)
intrepid (ANL IBM BG/P, os is BGP, 4 pes/node, batch system is cobalt)
jaguar (ORNL XT4, os is CNL, 4 pes/node, batch system is PBS)
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

- The component and component models are basic element throughout CESM
- Plug and play of components (ie atm) with different component models (ie cam, datm, etc)
 - Done at case configuration time
 - Each component model has its own sub-directory tree under the model root





(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

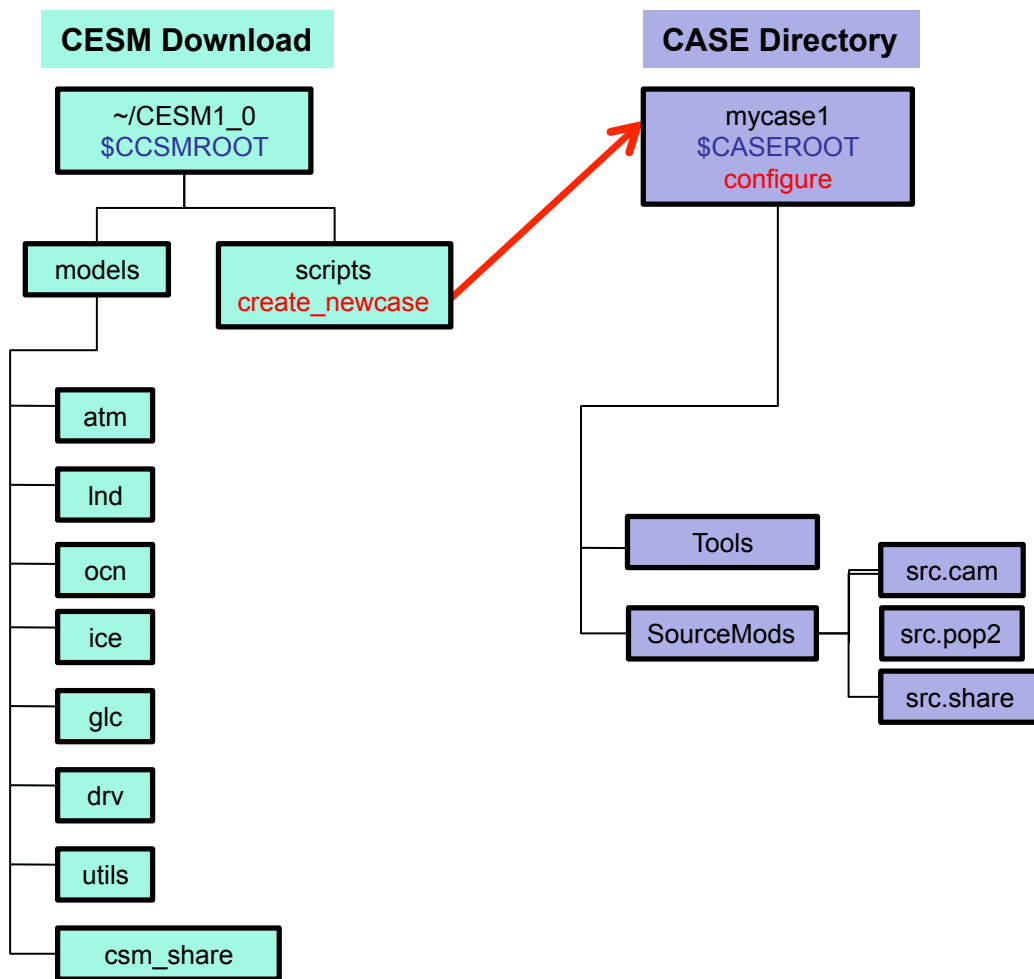
success



(1) Overview of Directories (+ create_newcase)

INPUTDATA Directory

/fs/cgd/csm/inputdata
\$DIN_LOC_ROOT_CSMDATA





(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,nstep,eps,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" />

<!--"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**
 - set number of component mpi tasks and number of component openmp threads, and 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" />
```



Next Step In the 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) Running the Model: Initial Run and Output Data
 - (5) Running the Model: Continuation Runs



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

- Step (2) is to “configure” the case using the configure script
- 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*, not after
- `configure -clean` unlocks the above files

```
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*
 - Generates Buildconf/ directory and buildnml, buildexe, and input_data_list files
 - Generates the case .build and .run scripts
 - Locks env_conf.xml and env_mach_pes.xml
- *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

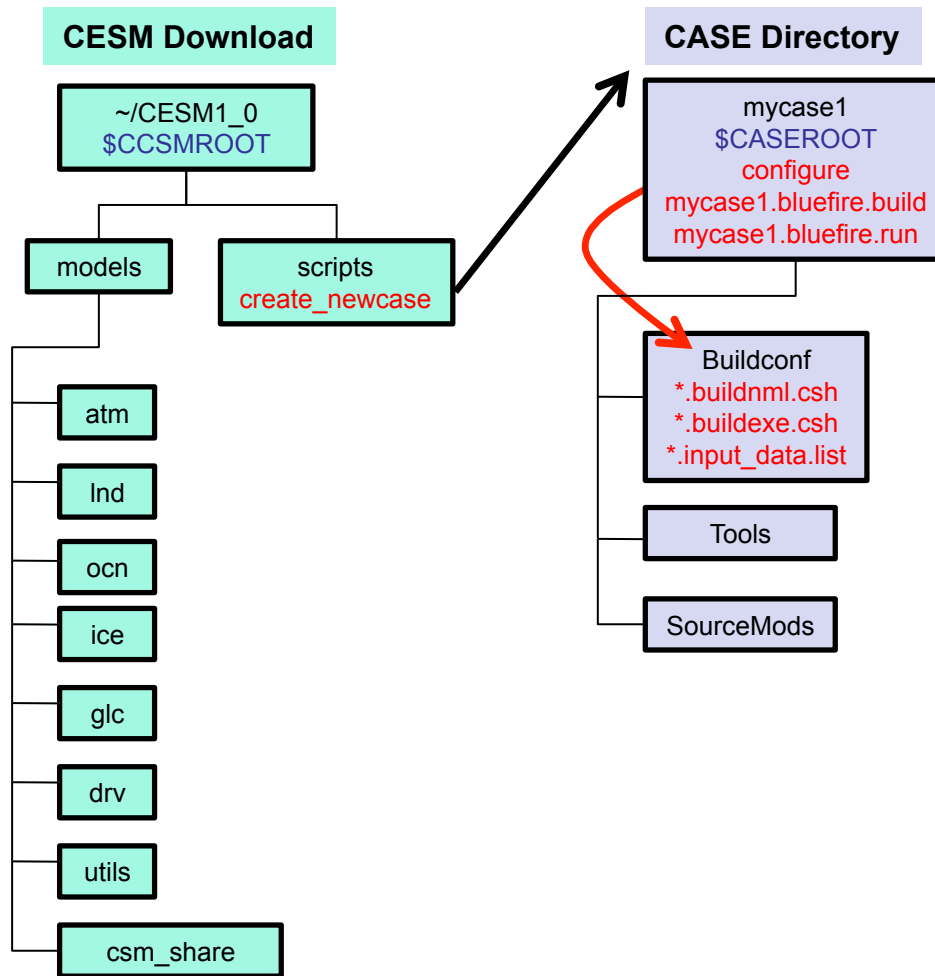
Success



(2) Overview of Directories (+ configure)

INPUTDATA Directory

```
/fs/cgd/csm/inputdata  
$DIN_LOC_ROOT_CSMDATA
```





(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
```



Next Step In the 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 Runs



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, not after
 - 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 that will contain executable code and model namelist files
 - locks env_build.xml
 - builds the individual component libraries and model executable
- **If any inputdata is missing...**
 - build will abort, but it will provide a list of missing files
 - must run **check_input_data -export** to acquire missing data
 - this will use svn to put required data in the inputdata directory
 - must re-run build script after running **check_input_data -export**



(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

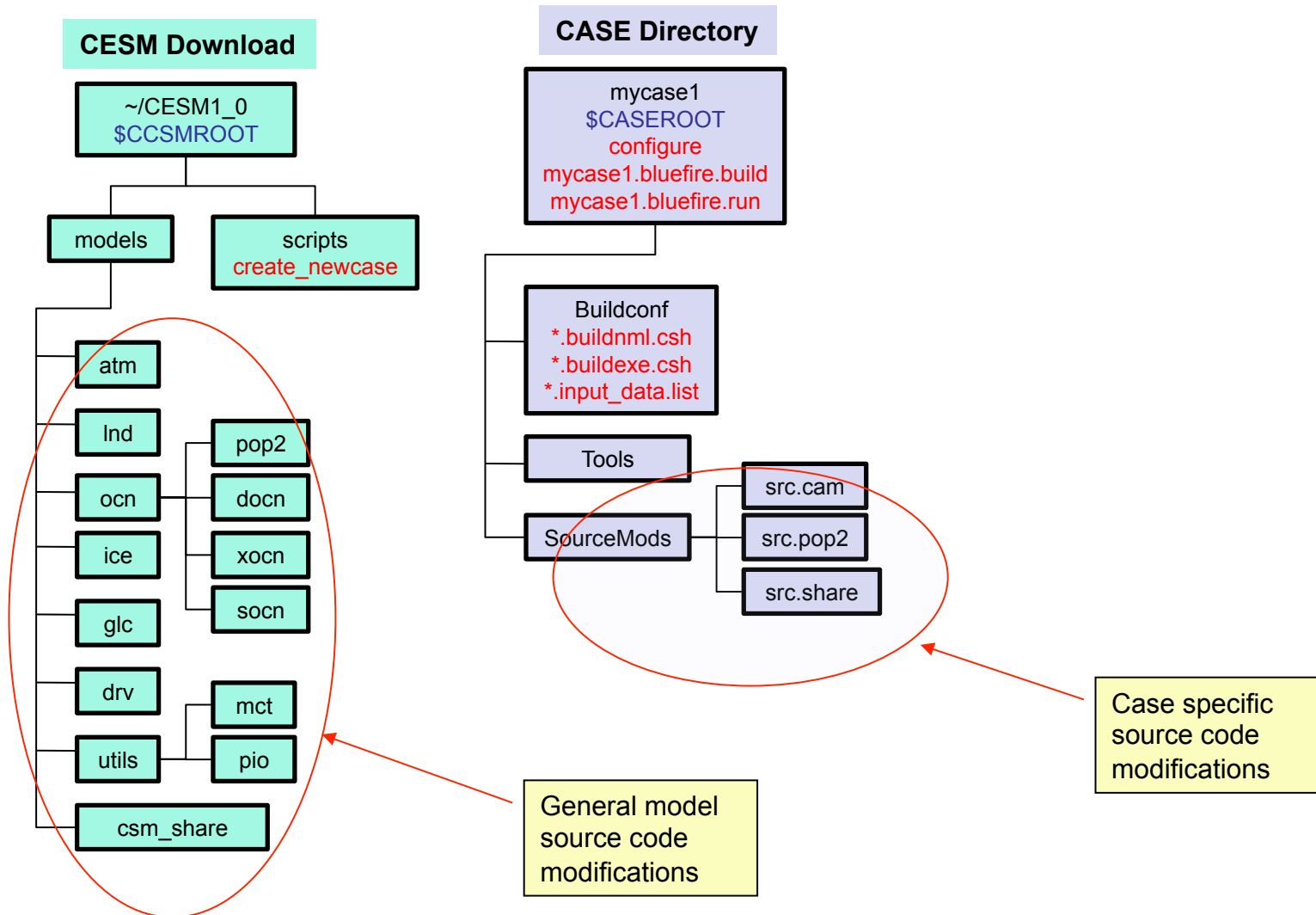
env_build.xml

.build script



(3) Modifying Source Code

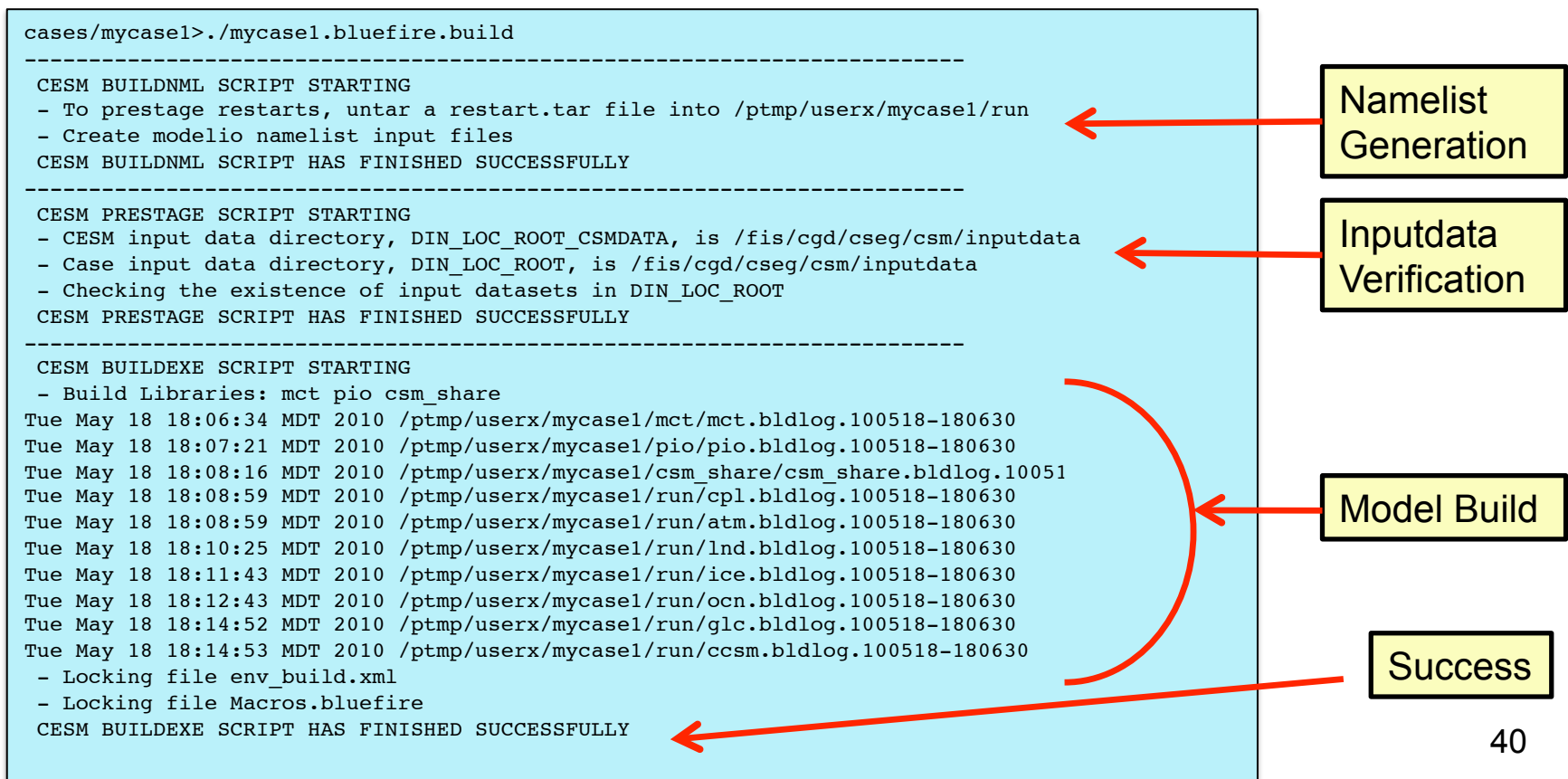
- Code modified in the models directory will apply to all new cases created – A BAD IDEA
- Modified code in the CASE SourceMods subdirectory applies to that case only
- Files in the SourceMods/ must be in proper subdirectory, eg. pop2 code must be in src.pop2





(3) Running the .build Script

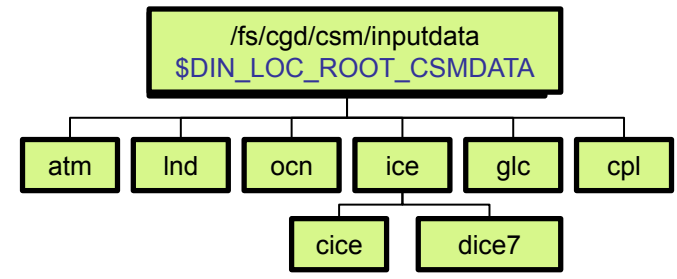
- Checks for missing input data
- Aborts if any input data is missing
- Builds the component model libraries and executable by running the *.buildexe.csh scripts for each component



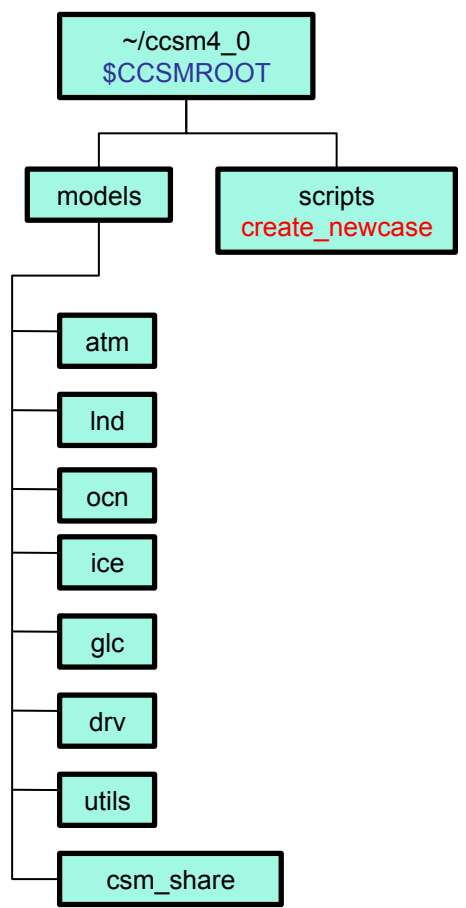


(3) Overview of Directories (+ build)

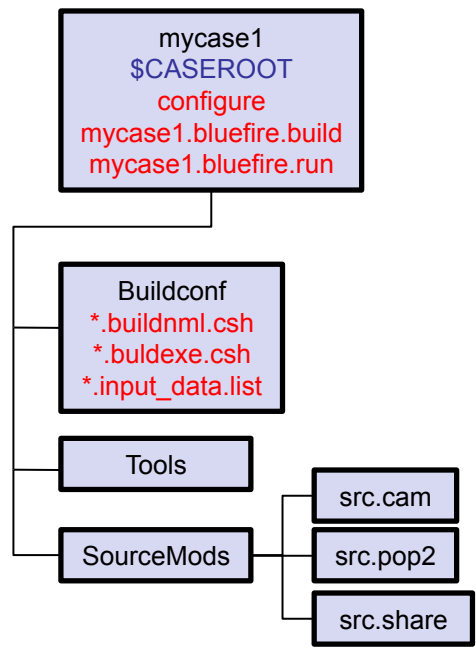
INPUTDATA Directory



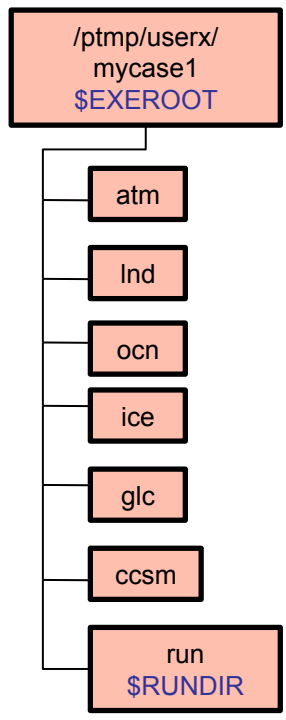
CCSM Download



CASE Directory



Build/Run Directory



(3) Your \$RUNDIR after running .build

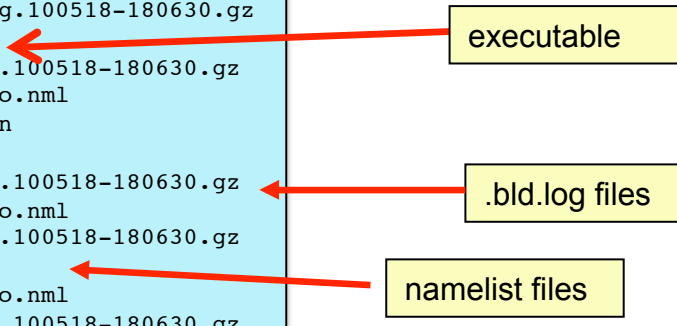
```
cases/mycase1> source Tools/ccsm_getenv
```

```
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
```





Next Step In the 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 Runs



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/cesml_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: an Initial Run

- **Step (4) – do an initial run (default is 5 days)**
- **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  be1105en  be1105en  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 ccsml.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 ccsml_timing.mycase1.100518-183212
-rw-r--r-- 1 userx ncar 3711 May 18 18:37 ccsml_timing_summary.100518-183212.gz
```

Copies of the Current Namelist Input Files

stdout/err

Log Files

Timing Files



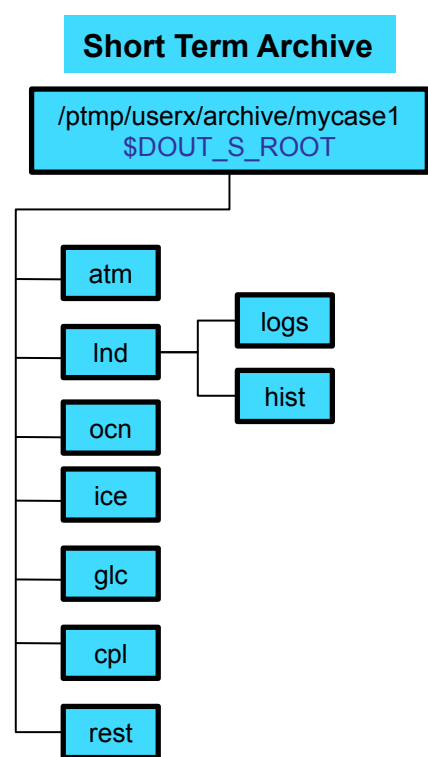
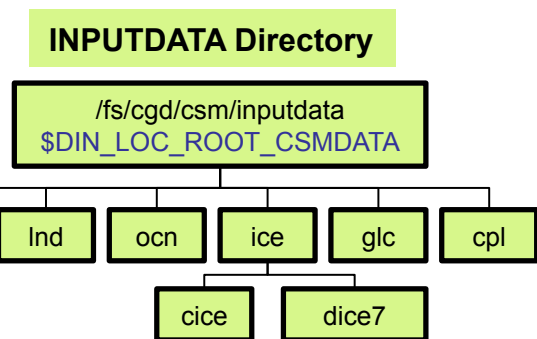
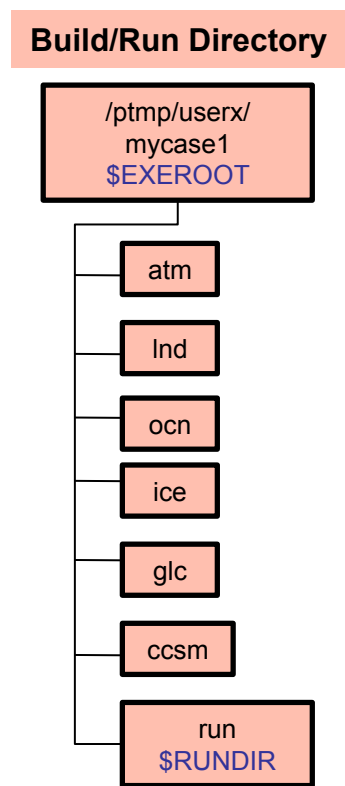
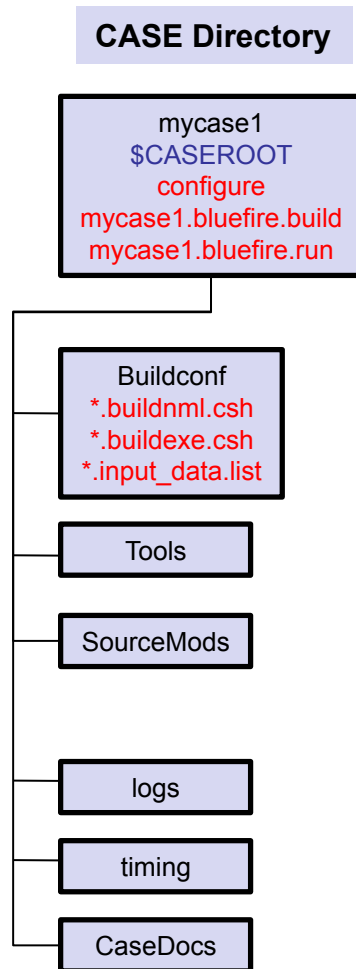
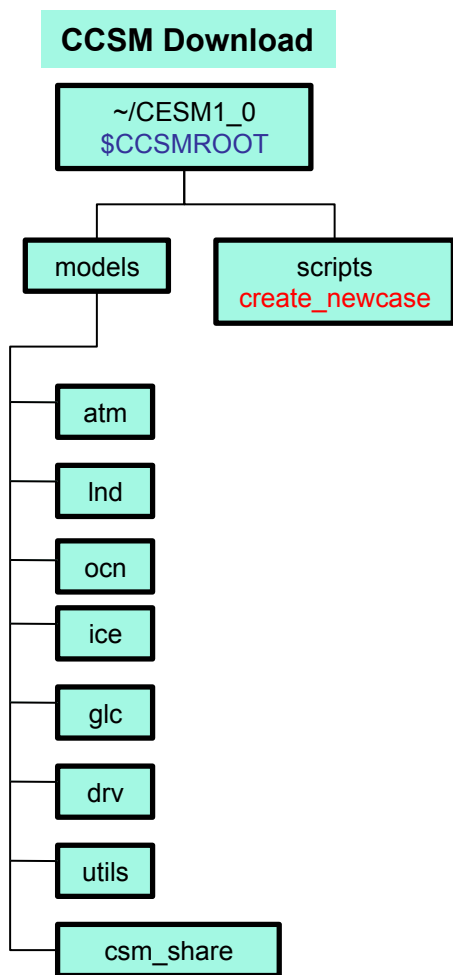
(4) Output in Short Term Archiving Directory

- Output data is originally created in ***\$RUNDIR***
- When the run ends, output data is moved into a short term archiving directory, ***\$DOUT_S_ROOT***
- **Why?**
 - Cleans up the ***\$RUNDIR*** directory
 - Migrates output data away from a possibly volatile ***\$RUNDIR***
 - Gathers data for the long term archive script which can then save the data to a permanent long-term storage area (e.g. HPSS)

```
cases/mycase1>echo $DOUT_S_ROOT
/tmp/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)





Next Step In the 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) Running the Model: Initial Run and Output Data
 - (5) Running the Model: Continuation Runs**



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**
 - **Probably 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)**

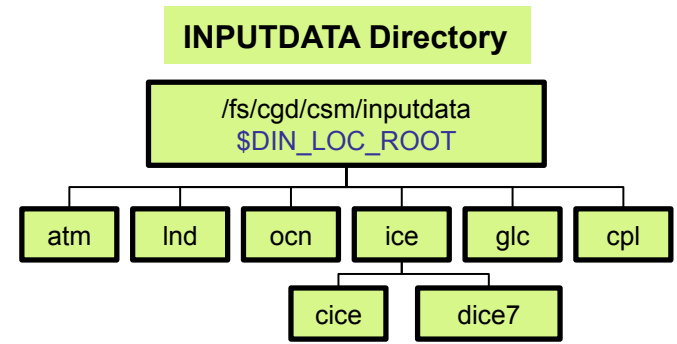
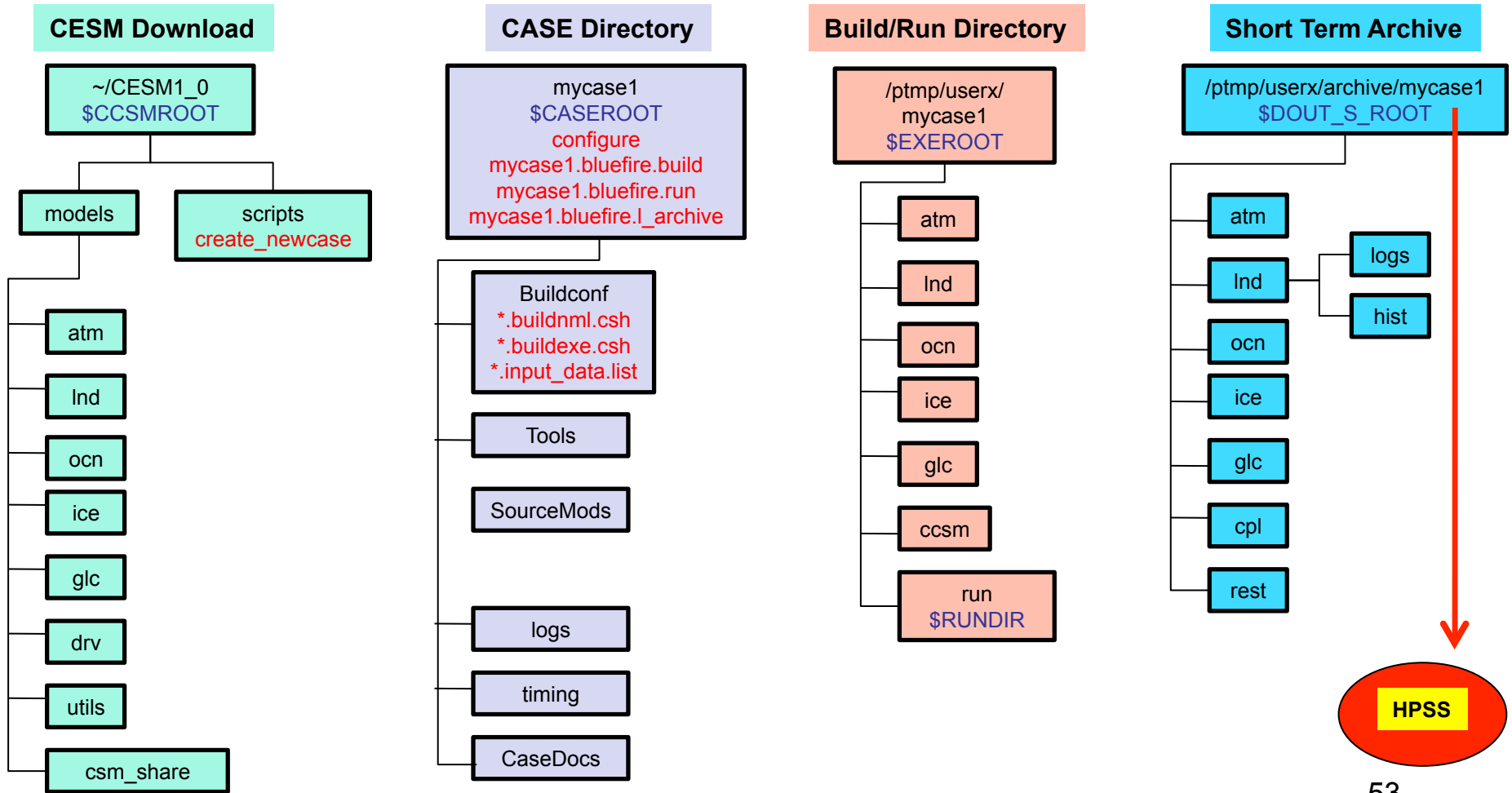


(5) Long Term Archiving

- **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` in `env_run.xml` (and also 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 with a production job or vice versa.



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





More Information/Getting Help

- **Model User Guides (please provide feedback)**
 - <http://www.cesm.ucar.edu/models/cesm1.0.4/>
 - CISM 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.



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 move 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**

```
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 =====
```

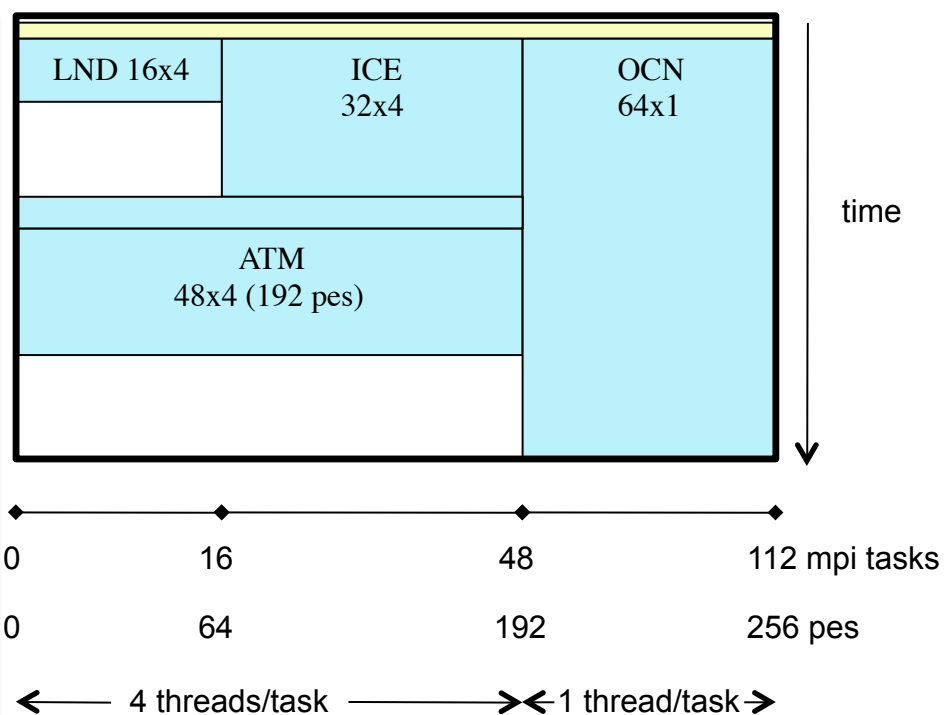


Appendix E (cont): Performance & Load Balance

- **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

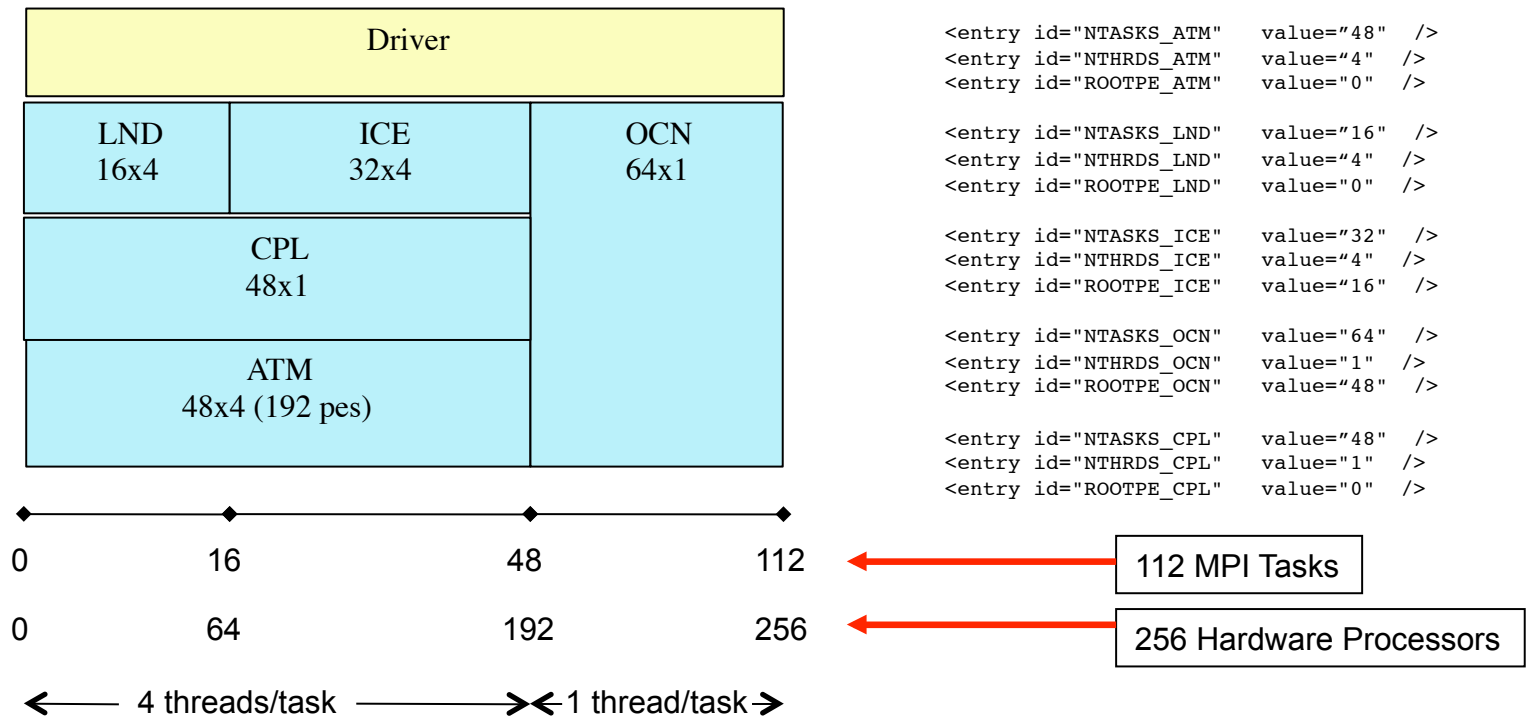




Appendix E (cont): Load Balancing & env_mach_pes.xml

- 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