CCSM4 / CESM1 Tutorial

CCSM Software Engineering Group

Updated July 7, 2010

CCSM 4.0 release, April 1, 2010

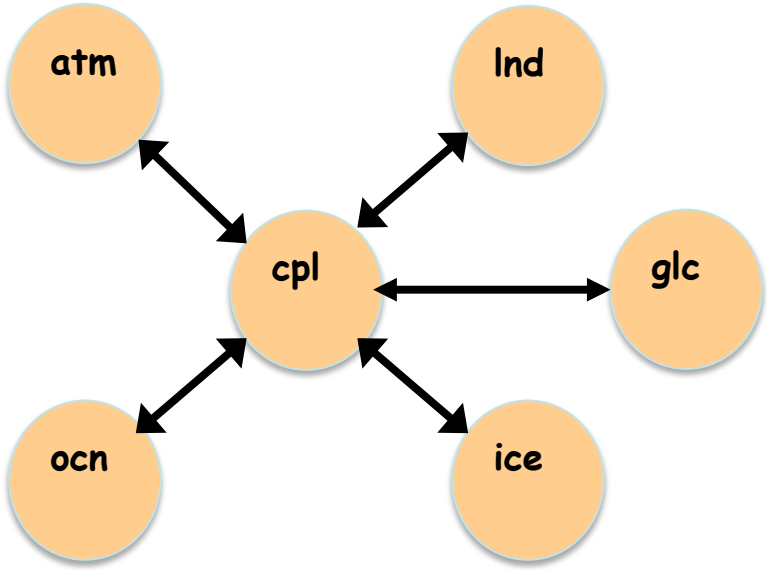
CESM 1.0 release June 25, 2010

Outline

- What is CESM?
- 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 Outuput Data
  - Continuation Runs
- Getting More Help
- Appendix

What is CESM?

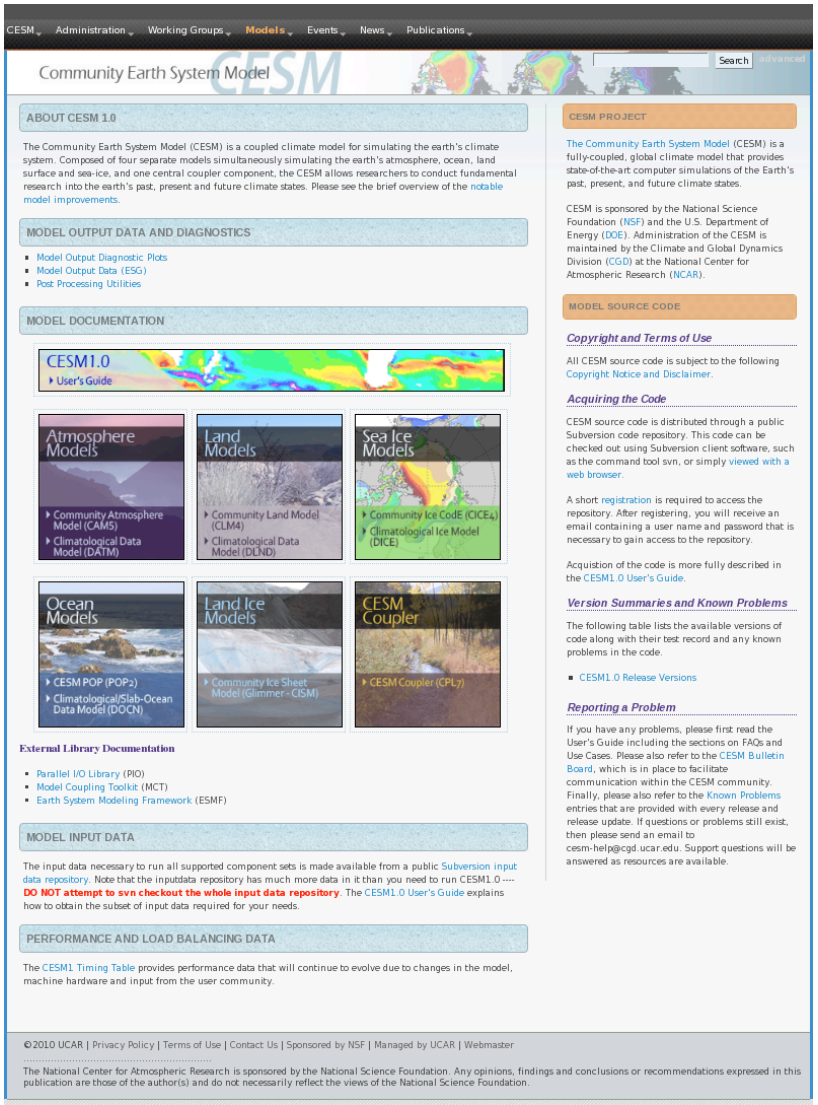
- Community Earth System Model (CESM) is a coupled climate model used for basic climate research, contributions to IPCC reports, and hindcasts and forecasts of past, present, and future climates.
- Consists of 4+ geophysical component models (atmosphere, land, ocean, sea ice) *on potentially different grids* that exchange *boundary data* with each other ***only*** via communication with a coupler (hub and spoke architecture)



- The component is a basic element throughout CESM
  - Directory Structures
  - Plug and Play Components (ie atm) with different Models (ie cam, datm, etc)
  - Configuration Setup

Release Homepage on Web

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



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 netcdf3.6.x)
- 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
  - jaguar and jaguarpf – ORNL Cray XT4 and XT5
  - franklin and hopper – NERSC Cray XT4 and XT5
  - kraken – NICS Cray XT5
  - intrepid – ANL IBM Bluegene/P
  - edinburgh – NCAR linux cluster
  - hadley – LBL linux cluster
  - midnight – ARSC Sun cluster
  - brutus – ETH linux cluster with pgi/openmpi, pgi/mpich, intel/openmpi, or intel/mpich
  - and a few others

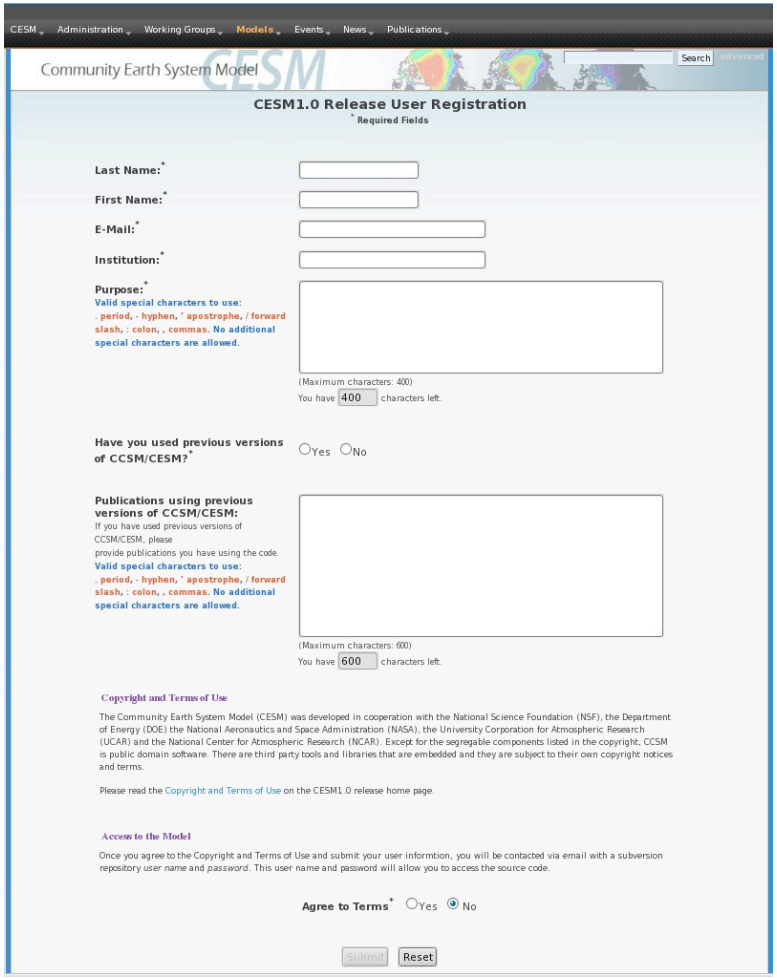
Basic Work Flow

- 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

(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 to access the repository



(A) Download the Source Code

- Code and input datasets are in a subversion repository (get subversion at http://subversion.apache.org/)
- Do NOT download any 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/
/
```

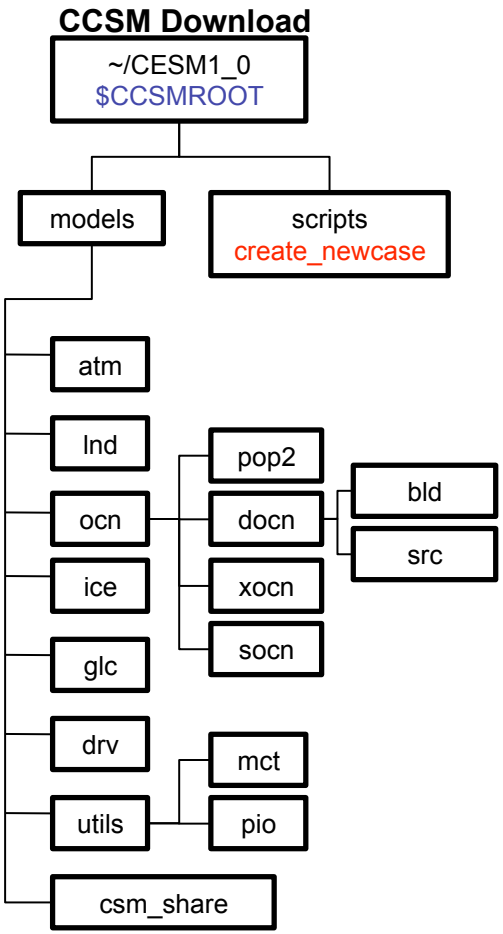
login

password

> svn co --username guestuser https://svn-ccsm-release.cgd.ucar.edu/models_versions/cesm1_0

```
>svn co --username guestuser https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_0
A    cesm1_0/models
A    cesm1_0/models/dead_share
A    cesm1_0/models/dead_share/dead_data_mod.F90
A    cesm1_0/models/dead_share/dead_mod.F90
A    cesm1_0/models/dead_share/dead_mct_mod.F90
A    cesm1_0/models/ocn
A    cesm1_0/models/ocn/pop2
 …
```

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

(B) Create an Inputdata Root Directory

- The inputdata area contains the input data required to run the model
- On supported machines the inputdata directory already exists with all necessary data
- Location specified in the scripts by the $DIN_LOC_ROOT_CSMDATA variable in env_run.xml

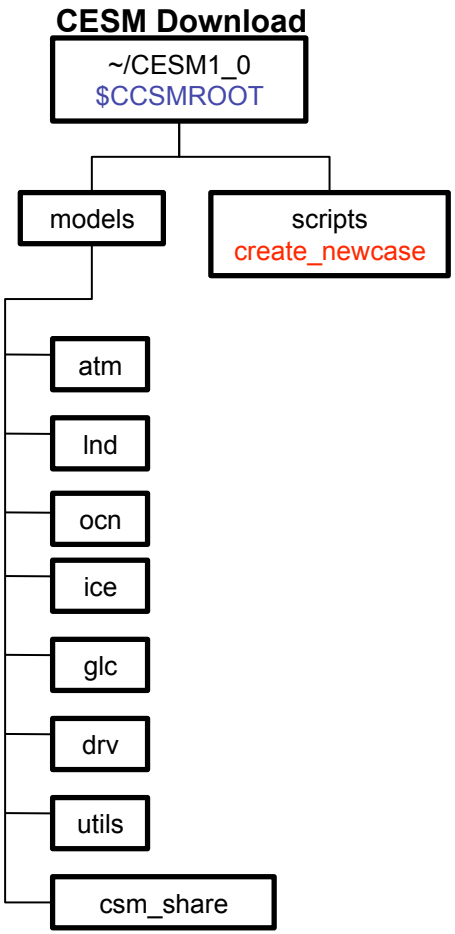
- On non-supported machines you will need to create an inputdata root directory and add the data
- Ideally this directory is shared by a group of users to save disc space
- Initially the inputdata directory contains no data, it will be added later on an as-needed basis
- Do NOT download input data manually (ie. by using svn co)
- The script `check_input_data` is used to download data into the inputdata
- The script `check_input_data`…
  - checks if the necessary data is already available in the inputdata dirctory
  - downloads only the data needed for a particular run (more later)
  - puts the data in the proper subdirectories
  - 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



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

(C) Porting

- Porting is beyond the scope of this presentation

- On supported machines, no porting is necessary
- See the User's Guide for a list of supported machines
- See the User's Guide for help on porting to a new machine
- See the Appendix for a few tips on how to port
- If the new machine is similar to a supported machine, porting can be relatively easy
- Porting might also be quite difficult – a lot depends on the specifics of your machine

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
```

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

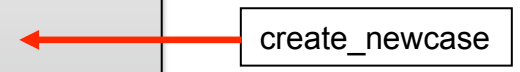
These unix commands built and ran the model on a supported machine named "bluefire"

```
# 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
```

(1) Create a New Case

- First step in setting up a model run is running the script `create_newcase`
- Go to the scripts directory: …/CESM1_0/scripts/
- `create_newcase` is the tool that generates a new CESM1 case
- Scripts are a combination of csh, perl, sh, and xml.

```
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
```



(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 settings) (optional).
     -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>           ccsm 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_CN -mach bluefire

- "case" is the name and location of the case being created (eg. ~/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", ie. which component models to use
  - Eg. "B" compsets use all active models
  - Eg. "F" compsets use active atmosphere, land, and ice, but data ocean model
  - Equivalent short and long names (B1850CN == B_1850_CN)

- "mach" specifies the machine that will be used.
  - "supported" machines like bluefire, jaguar, franklin, intrepid
  - "prototype" machines that are not tested regularly, eg. prototype_frost
  - "generic machines" provide a starting point for porting, eg."generic_ibm

- Create_newcase -list will list all the valid choices for these command line options (see next slide)
- Once these values are set on the command line, they cannot be changed, they are "locked down"
- The file env_case.xml contains all the things that were "locked down" 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)
```

(1) Running create_newcase

```
CESM1_0/scripts>./create_newcase -case ~/cases/mycase1 -res f19_g16 -compset B -mach bluefire

*********************************************************
* CCSM has significant flexibility to configure cases
* with respect to components, grids, and model settings.
* This version of CCSM has only be validated scientifically
* for the following full active configurations:
*
*  1.9x2.5_gx1v6   B_1850_CN
*  1.9x2.5_gx1v6   B_1850_RAMPCO2_CN
*  0.9x1.25_gx1v6  B_1850_CN
*  0.9x1.25_gx1v6  B_1850_RAMPCO2_CN
*
* If the user is interested in running a "stand-alone"
* component configuration, the following model
* configurations have been validated scientifically
* have associated diagnostic output as part of the release.
*
*  0.9x1.25_gx1v6  F_AMIP_1DEG
*  1.9x2.5_gx1v6   F_AMIP_2DEG
*  0.9x1.25_gx1v6  I and ICN
*  T62_gx1v6       C
*
* Please refer to the individual component release
* web pages for information regarding alternative
* component configurations.
*********************************************************

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

Creating /blhome/userx/cases/mycase1

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

warning message

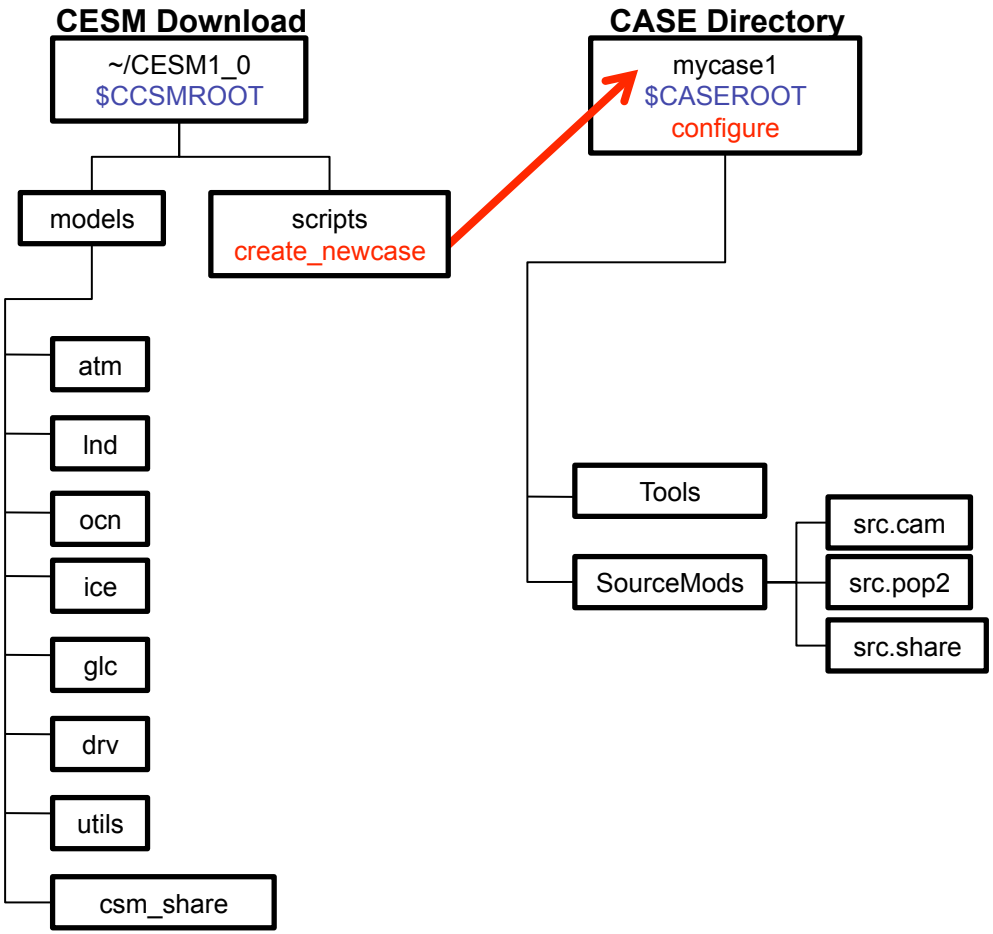
case location

success

(1) Overview of Directories (+ create_newcase)

INPUTDATA Directory

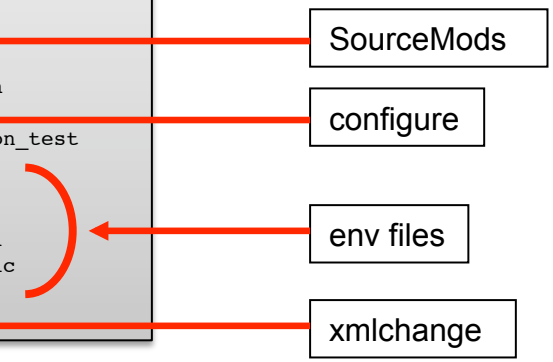
/fs/cgd/csm/inputdata
$DIN_LOC_ROOT



(1) Case Directory After Running create_newcase

- SourceMods is a directory where case specific code modification can be placed
- configure is the script used in the next step, step (2)
- env_*.xml contain environment variables associate with the case (more on this later)
- xmlchange is a script that changes env variable values through a command line interface

```
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
```



(*) About .xml Files: Format & Variables

- Contain 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" is the variable name
- "value" is the variables name's setting
- <!--- text --> is a comment in xml
- Most variables have a description (above the entry)
- "(type)" is the type of the variable
- "valid values" indicates the full set of allowable settings
  - the scripts will let you know if you try to set a variable to an invalid value
  - many values do not have valid values defined, that means there are no constraints
- Use `xmlchange` to modify env variable settings

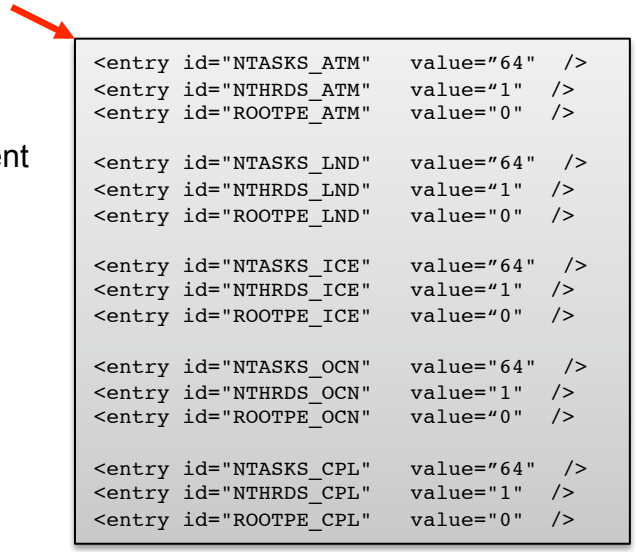
  > `xmlchange -help`

  > `xmlchange -file env_run.xml -id STOP_N -val 20`
- Or edit env_*.xml file manually -- but be careful about introducing formatting errors
- Use source to set your local environment with the case variable settings

  > `source ./Tools/ccsm_getenv`

(*) About .xml Files: How They Change the Build and Run

- Defaults are generally reasonable
- **env_case.xml** is set by create_newcase and cannot be modified
- **env_conf.xml** variables specify various component information
  - Most often this file should not be modified
  - RUN_TYPE, RUN_STARTDATE, RUN_REFCASE, RUN_REFDATE – defines initial conditions
  - Can change the physics of a model – be very careful about this
- **env_mach_pes.xml** variables specify the layout of components on hardware processors
  - Is used to tune the performance of the model
  - Scientific results do not depend on component/processor layout
  - 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
- **env_build.xml** variables specify some build information
  - Most often this file should not be modified
- **Macros.*** specifies the compilation variables used in the Makefile
  - Most often this file should not be modified
  - Can change compiler options, libraries, etc.
- **env_run.xml** variables specify run time information
  - Most often this file *will be* modified
  - STOP_OPTION, STOP_N, REST_OPTION, REST_N



```
<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

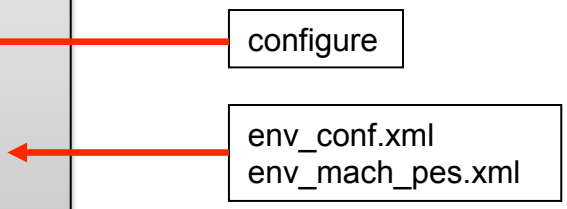
These unix commands built and ran the model on a supported machine named "bluefire"

```
# 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 ~/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
```

(2) Configure the Case

- Step (2) is to “configure” the case using the `configure` script
- Can modify env_conf.xml and env_mach_pes.xml before running `configure`, but not after
- Most often there is no need to modify env_conf.xml or env_mach_pes.xml
- Run `configure -case`
- Generates Buildconf directory with buildnml, buildexe, and input_data_list files
- Generates the case .build and .run scripts
- Locks env_conf.xml and env_mach_pes.xml

```
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
```



(2) About configure

> configure -help

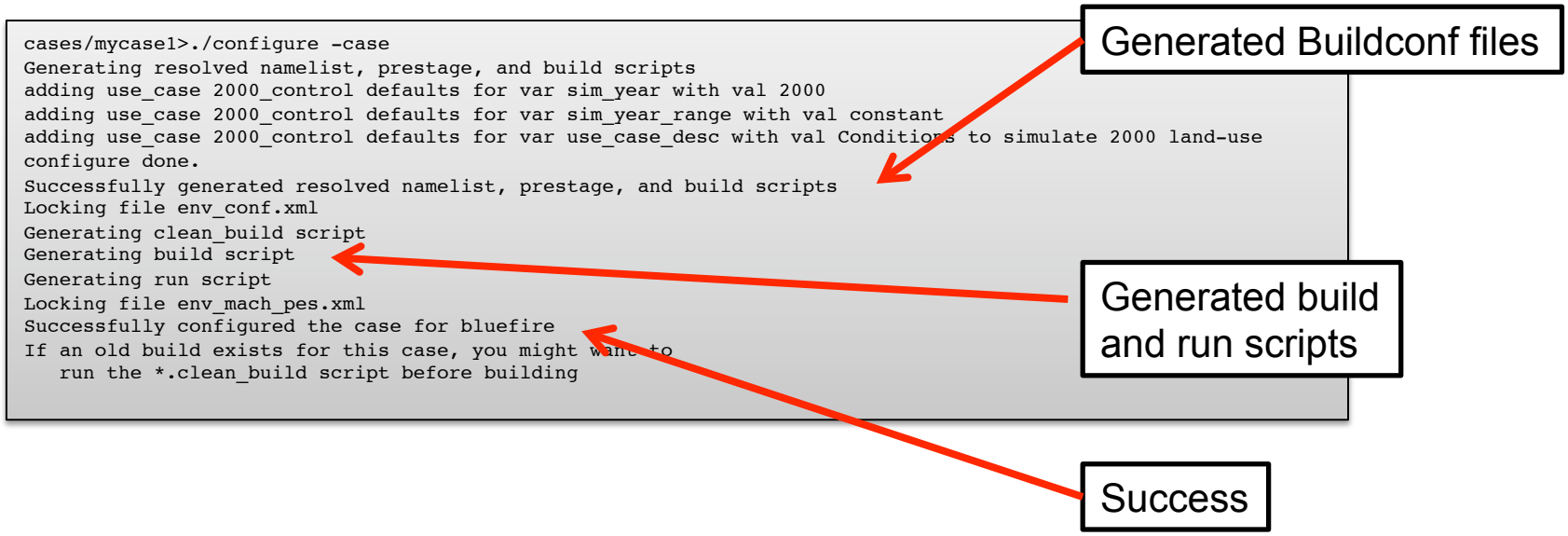
```
NAME

      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
  - Ban modify env_conf.xml and env_mach_pes.xml and type configure –case again

(2) Running configure

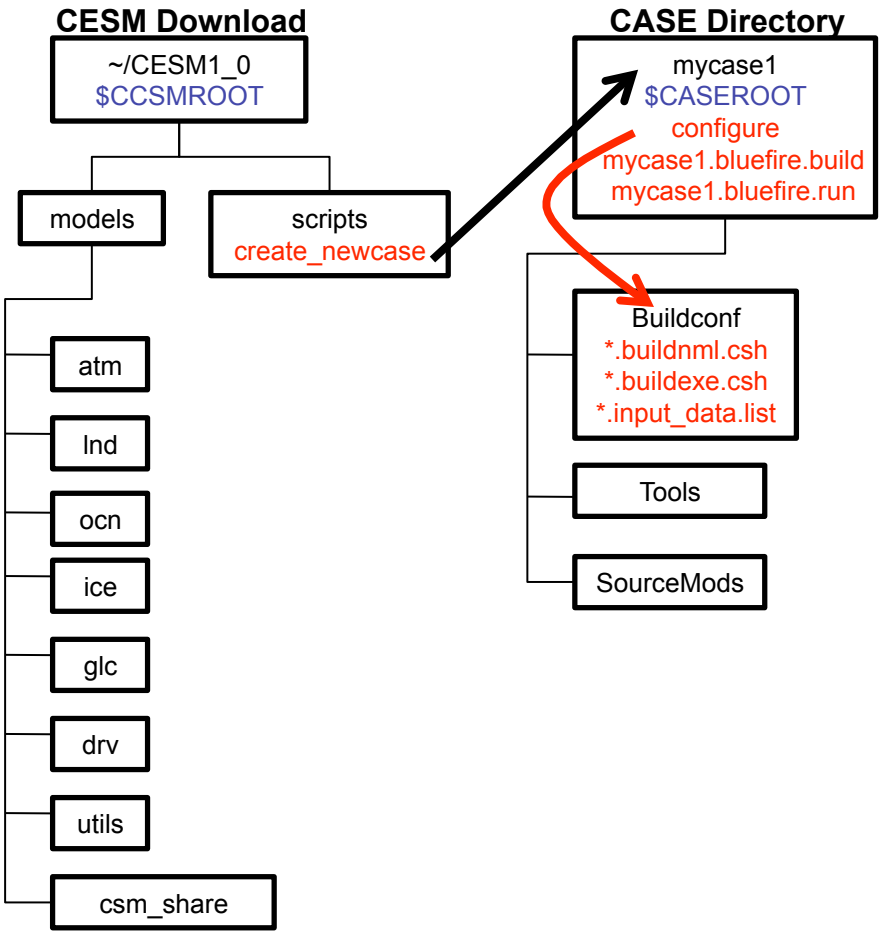
```
cases/mycase1>./configure -case
Generating resolved namelist, prestage, and build scripts
adding use_case 2000_control defaults for var sim_year with val 2000
adding use_case 2000_control defaults for var sim_year_range with val constant
adding use_case 2000_control defaults for var use_case_desc 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
```



(2) Overview of Directories (+ configure)

INPUTDATA Directory

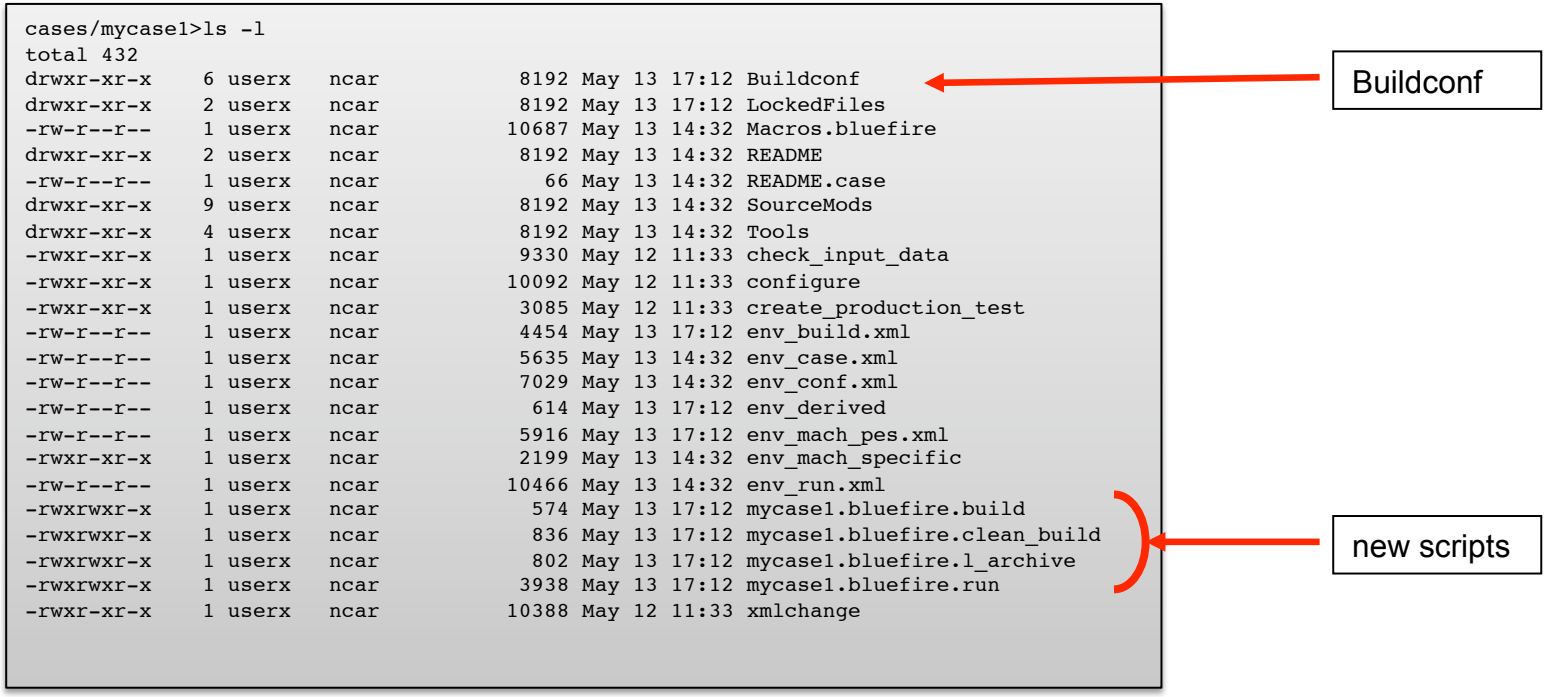
/fs/cgd/csm/inputdata
$DIN_LOC_ROOT



(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
```



(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 ccsm.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) 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
```

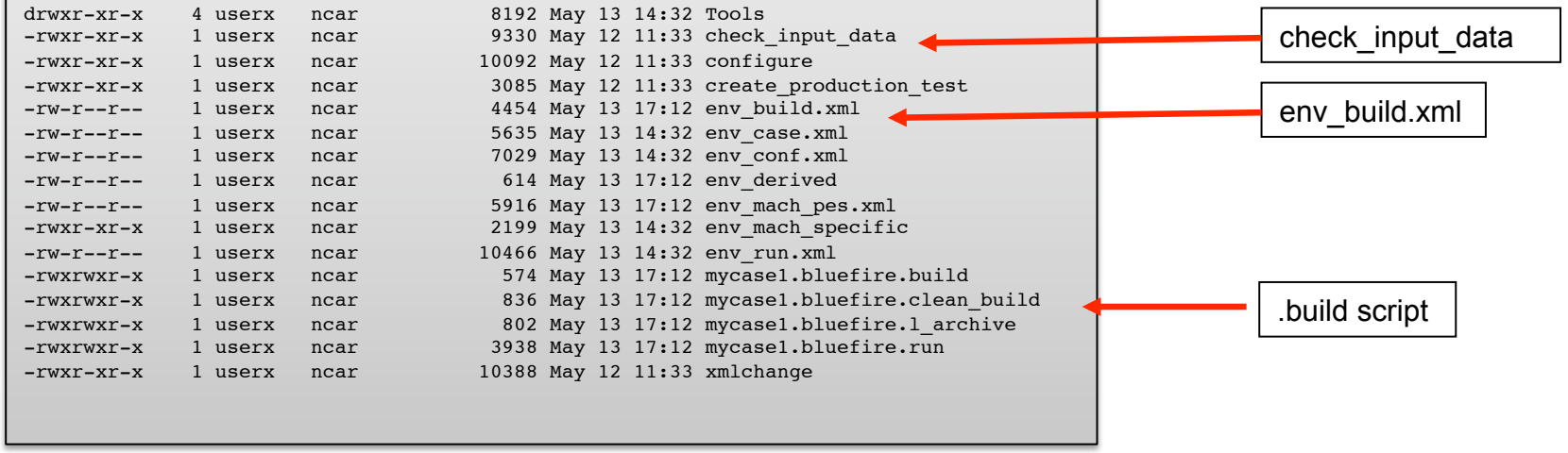
(3) Build the Model

- Step (3) is to “build” the model using the `*.build` script
- Can change values in env_build.xml before running *.build, but not after
- Most often there is no need to edit env_build.xml
- May want to introduce modified source code before building
- `.build` script creates build/run directory that will contain executable code and model namelist files
- Locks env_build.xml
- Builds the individual component libraries and then final executable

- Build script also checks for missing input data by running `check_input_data -check`
- If any inputdata is missing…
  - The build will abort, but it will provide a list of missing files
  - You must run the script `check_input_data –export` to acquire missing data
  - `check_input_data` will use svn to put required data in the inputdata directory
  - You must re-run build script after running `check_input_data`

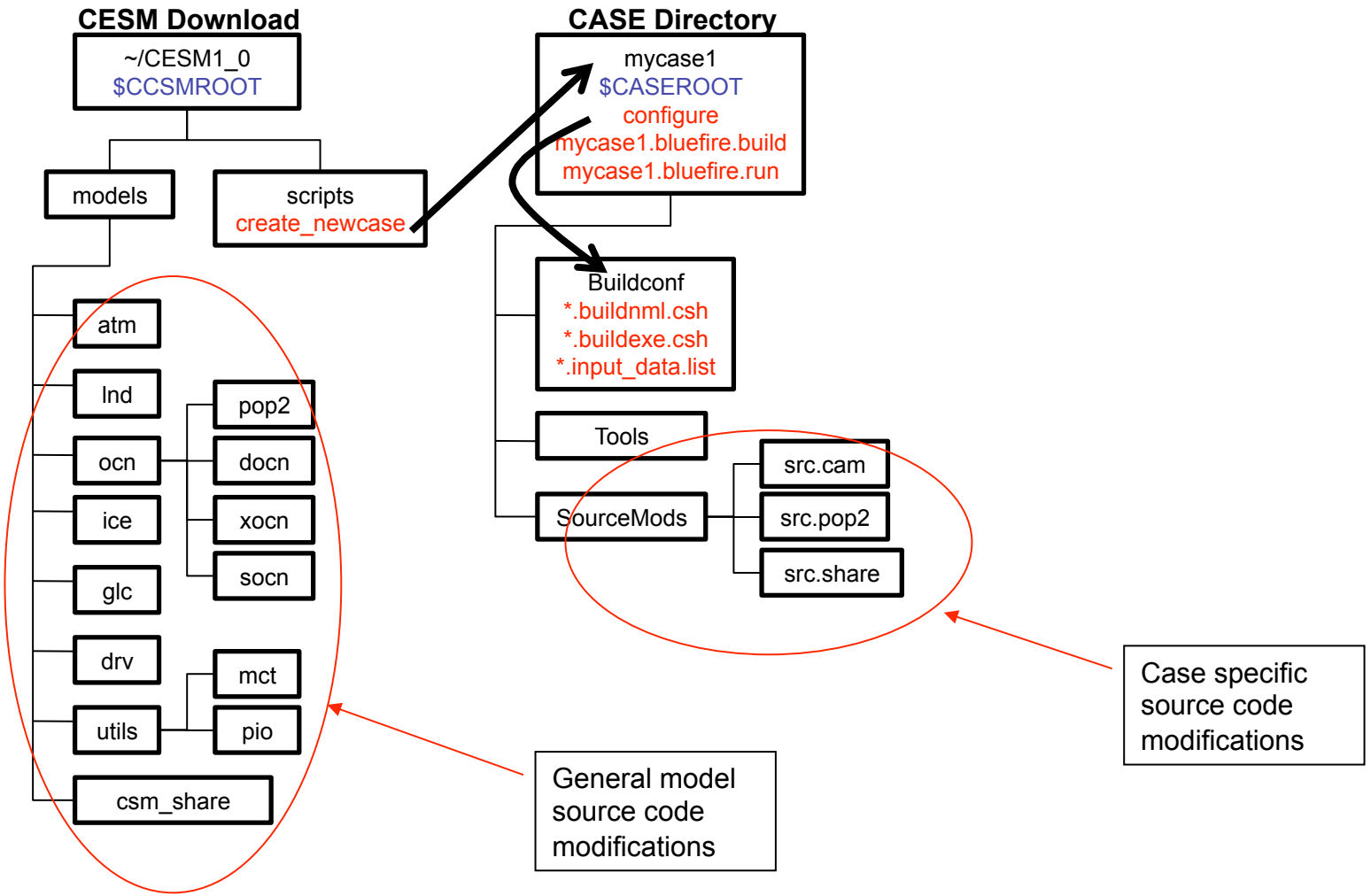
(3) The .build script

```
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
```



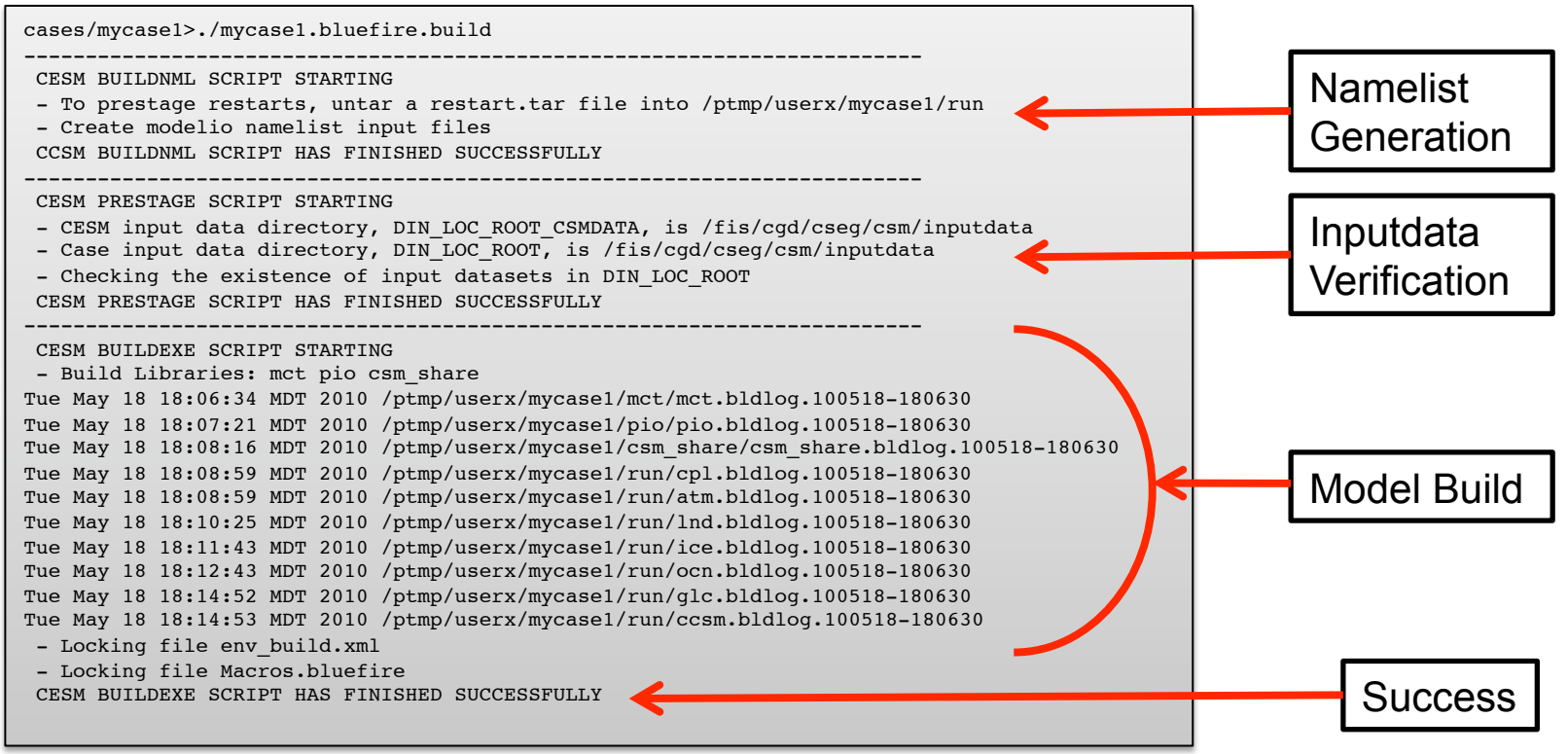
(3) Modifying Source Code

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

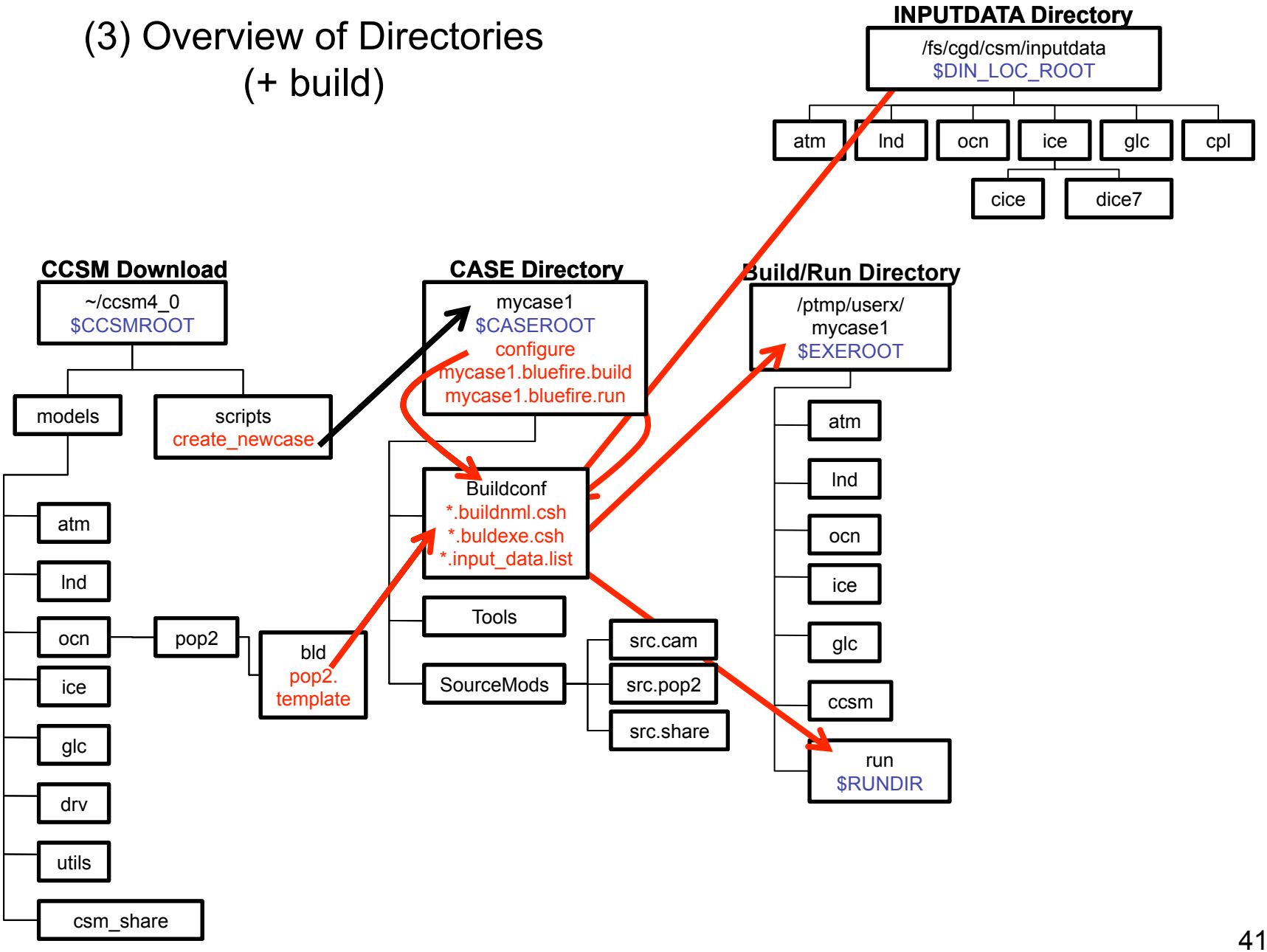


(3) Running the .build Script

- Generates input files (namelist) by running the buildnml.csh scripts for each component
- Checks for missing input data by running check_input_data
- If any inputdata is missing, the build will abort, but it will provide a list of missing files
- If any inputdata is missing, you must run the script check_input_data -export and then re-build
- check_input_data -export will use svn to put required data in the inputdata directory
- Builds the component model libraries and then final executable by running the buildexe.csh scripts for each component



(3) Overview of Directories (+ build)



(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
```

executable

.bld.log files

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
```

(4) Running the Model: an Initial Run

- Step (4) is to do an initial run of the model
- Initial runs are usually short (eg. 5-days) used to verify the model is running correctly
- May want to edit env_run.xml file before running (eg. to specify length of run)
- May want to modify namelist settings before running the model
  - Via env_run.xml variables
  - Directly in the Buildconf/*.buildnml.csh files
- The run script
  - Generates the namelist/input files (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                 mycase1   May 18 18:30
```

(4) Output in Your CASE Directory

- A job completed successfully if "`SUCCESSFUL TERMINATION OF CPL7-CCSM`" appears near the 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             0 May 18 17:55 ls
-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          3711 May 18 18:37 ccsm_timing_summary.100518-183212.gz
```

Copies of the Current Namelist Input Files

stdout/err

Log Files

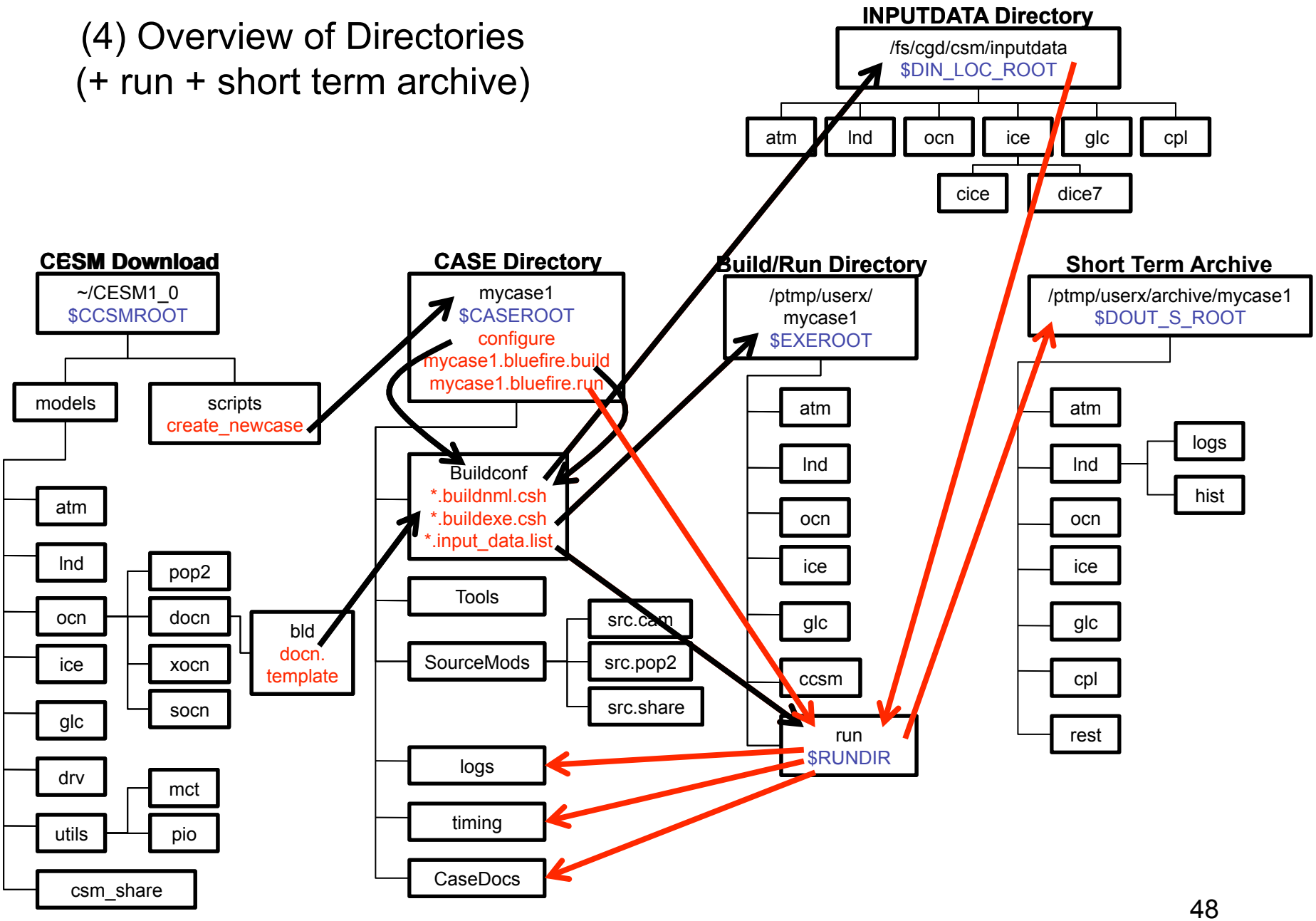
Timing Files

(4) Output in Short Term Archiving Directory

- Output data is originally created in $RUNDIR
- But when the run ends, output data is moved into a short term archiving directory
- Why?
  - Cleans up the $RUNDIR
  - Migrates output data away from a possibly volatile $RUNDIR
  - Gathers data for the long term archive script which can then save the data a permanent long-term storage area

```
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 ccsm.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

- You should start with a short initial run, described in step (4)
- Carefully examine the output to verify that the run is doing what you want it to
- You might rerun the initial run several times to fix problems
- If the initial run looks good…

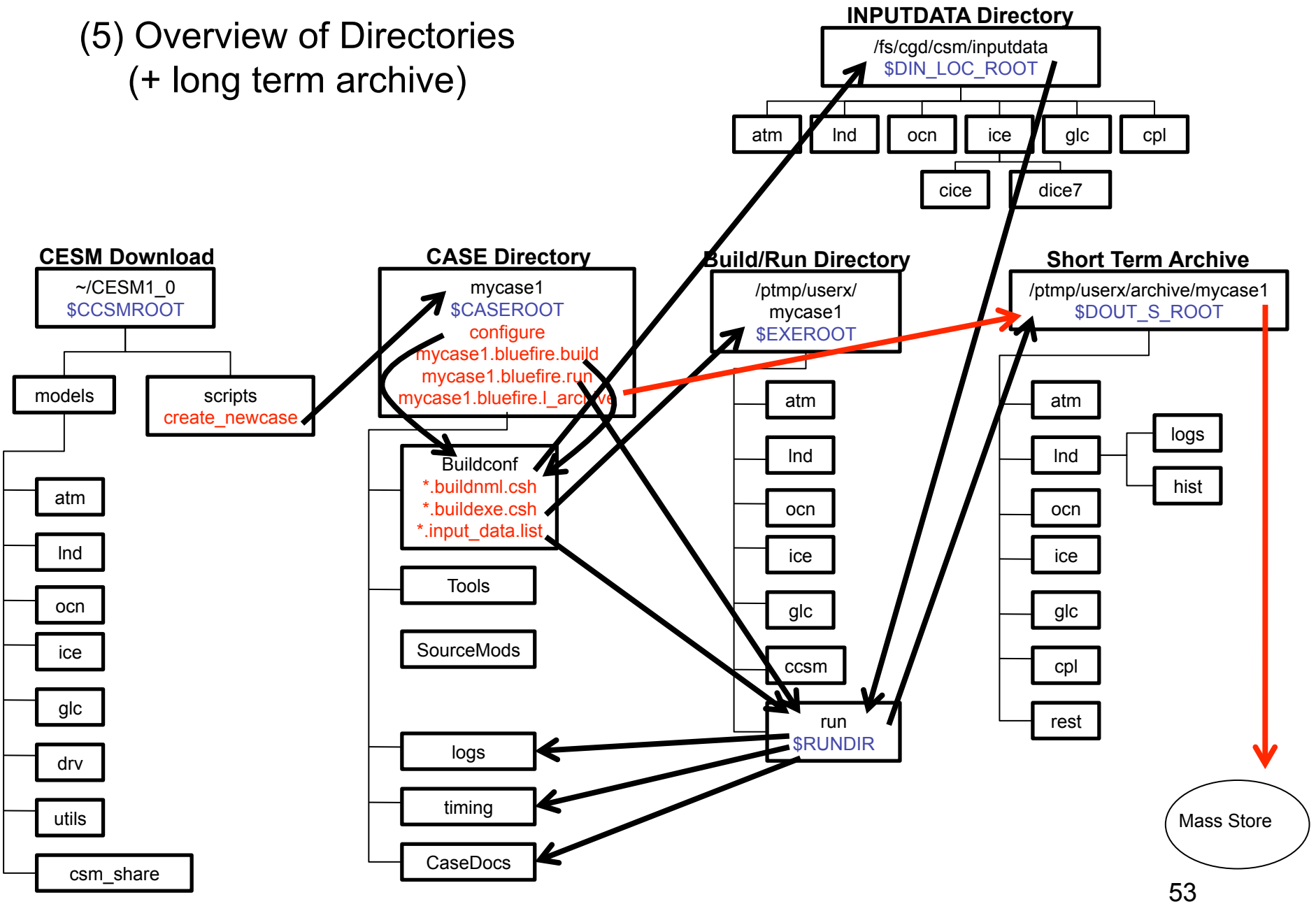
- Step (5) is a continuation run
- Change CONTINUE_RUN to TRUE in env_run.xml
- Probably will change STOP_OPTION to run the model longer
- May want to turn on auto-resubmit option in env_run.xml (RESUBMIT)
- May want to turn on “long term archiving” in env_run.xml (DOUT_L_MS)

(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; this 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.

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



More Information/Getting Help

- User Guides (http://www.cesm.ucar.edu/models/cesm1.0/) – please provide feedback
  - CESM Users Guide
  - CAM Users Guide
  - CLM Users Guide
  - POP2 Users Guide
  - CICE Users Guide
  - Data Models Users Guide
  - Coupler Users Guide

- 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.ccsm.ucar.edu/models/cesm1.0/tags/)
  - Notes significant bugs as they are identified

- Model output is available on the Earth System Grid (http://www.earthsystemgrid.org)

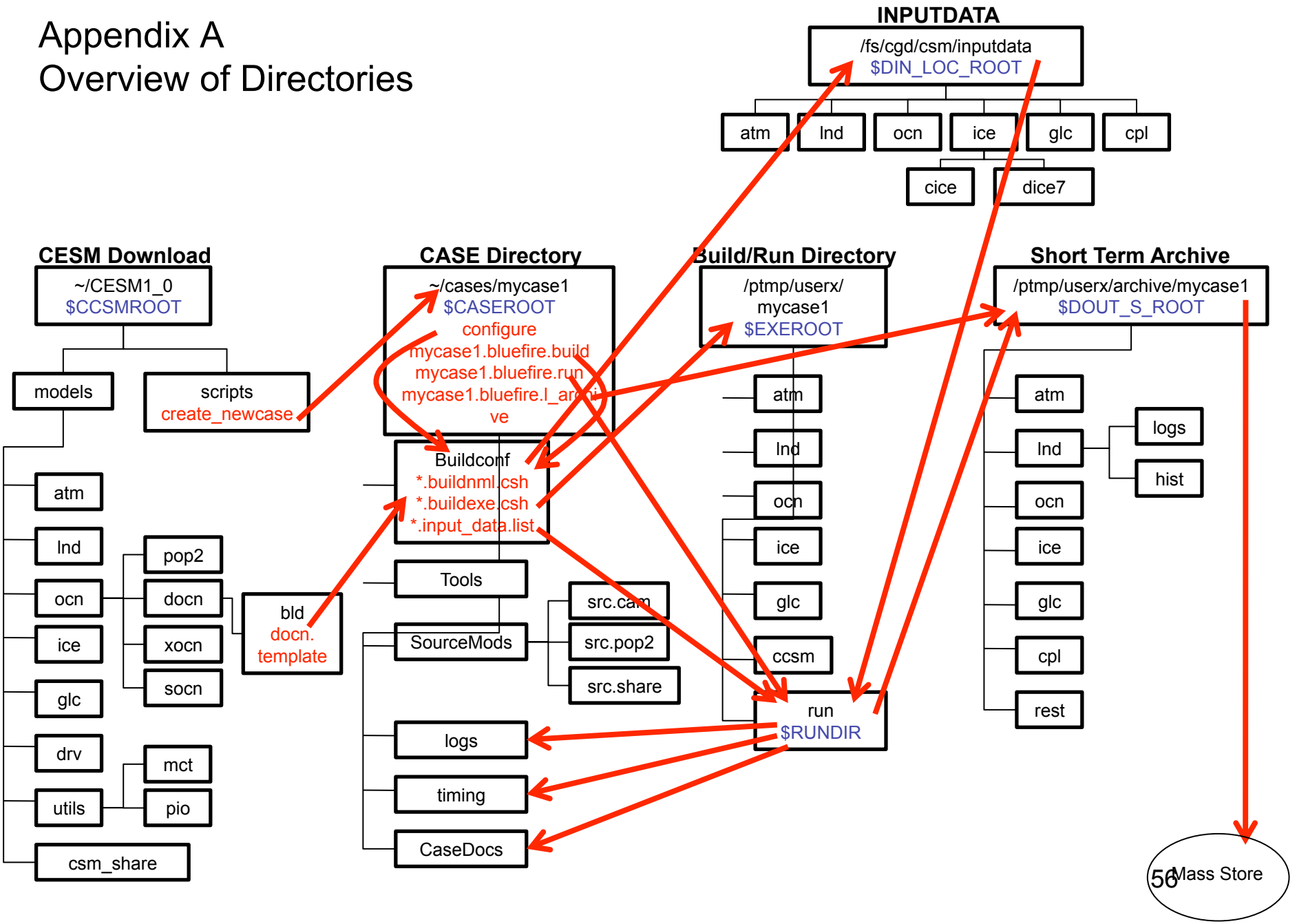
- Email cesm-help at cgd dot ucar dot edu
  - Questions will be answered as resources are available

The End

Appendix

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

Appendix A
Overview of Directories



Appendix B: 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 in env_run.xml	
$CASE*.l_archive	set DOUT_L to FALSE in env_run.xml	

Appendix C: Production Runs

- Verify configuration and inputs
- Verify performance, throughput, cost, and load balance
- Verify exact restart for the production configuration. Use “create_production_test” in the case directory.
- Carry out an initial run of a particular duration 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 the end of the run.
- When the 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
  - Set STOP_OPTION to “nmonths”, set STOP_N
  - Leave REST_OPTION==STOP_OPTION and REST_N==STOP_N to get a restart at the end of the run.
- 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; this 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 D: 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 E: 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 E: 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 E: 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 F: 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

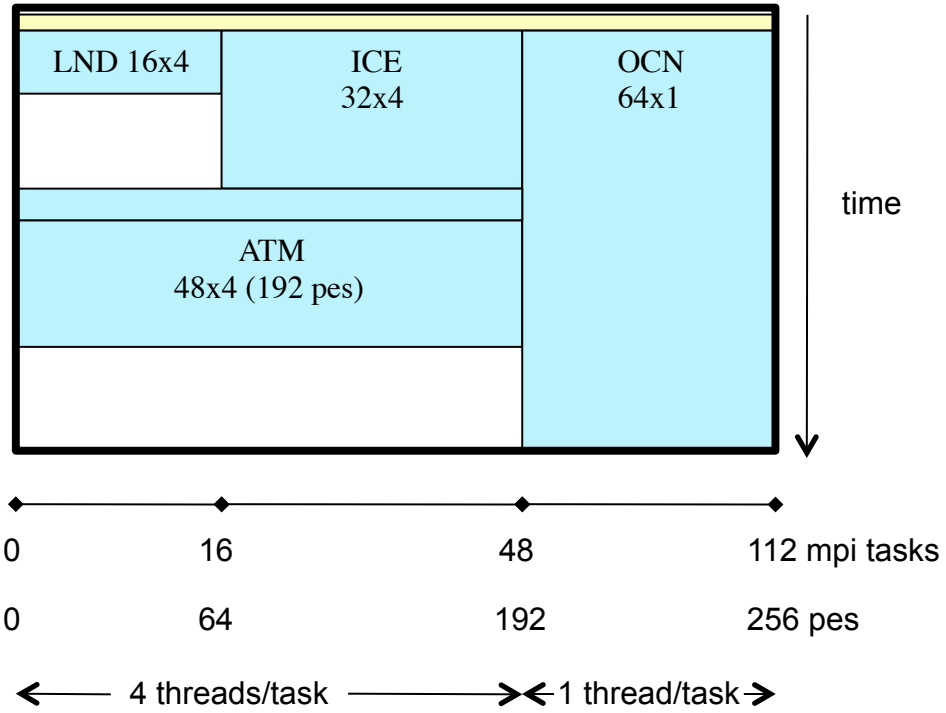
```
tStamp_write: model date =    10120       0 wall clock = 2010-05-19 21:11:07 avg dt =    16.43 dt =    16.12

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 F: 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

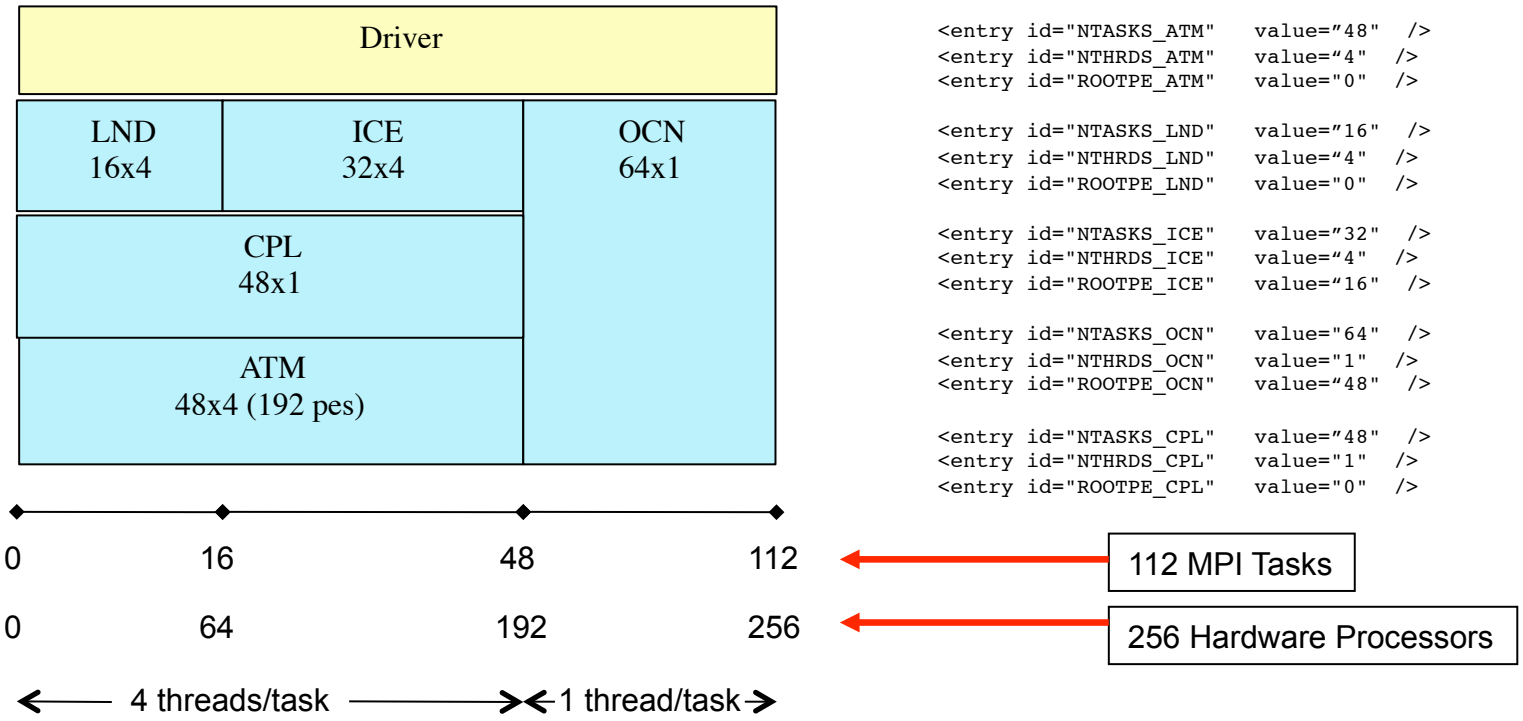
```
component       comp_pes    root_pe   tasks  x threads
 ---------       ------     -------   ------   ------
 cpl = cpl        48          0        48     x 1
 glc = sglc       64          0        64     x 1
 lnd = clm        64          0        16     x 4
 ice = cice       128         16       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 F: 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:




```
<entry id="NTASKS_ATM"    value="48"  />
<entry id="NTHRDS_ATM"    value="4"   />
<entry id="ROOTPE_ATM"    value="0"   />

<entry id="NTASKS_LND"    value="16"  />
<entry id="NTHRDS_LND"    value="4"   />
<entry id="ROOTPE_LND"    value="0"   />

<entry id="NTASKS_ICE"    value="32"  />
<entry id="NTHRDS_ICE"    value="4"   />
<entry id="ROOTPE_ICE"    value="16"  />

<entry id="NTASKS_OCN"    value="64"  />
<entry id="NTHRDS_OCN"    value="1"   />
<entry id="ROOTPE_OCN"    value="48"  />

<entry id="NTASKS_CPL"    value="48"  />
<entry id="NTHRDS_CPL"    value="1"   />
<entry id="ROOTPE_CPL"    value="0"   />
```

Appendix G: 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
```

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

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