CESM Tutorial

NCAR Earth System Laboratory
CESM Software Engineering Group

CESM 1.2.x and CESM1.1.x
CESM1.0.5 and previous (see earlier tutorials)

NCAR is sponsored by the National Science Foundation

Outline

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

CESM 1.2 Release Web Page

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

Release notes

Scientific validation

Post-Processing Tools

User's Guide

Component Model Documentation

External Libraries

Input Data

Timing Table

CESM1.2 PUBLIC RELEASE

ABOUT CESM 1.2

TO DO - general release description

CESM1.2 Release Notes includes What's New - Science, What's New - Software, Answer-Changing Features, Supported Machines, and Known Problems.

SCIENTIFIC VALIDATION

Scientific validation consists of a multi-decadal model run of the given component set at the target resolution, followed by scientific review of the model output diagnostics. All scientifically supported component sets are also accompanied by diagnostic and model output data.

CESM1.2 Scientifically Supported Model Configurations - CESM1.2 has the flexibility to configure cases with many different combinations of component models, grids, and models settings.

Validated CESM1.2 model results and diagnostics will be added to the site as they become available.

DIAGNOSTIC PACKAGES AND NAMING CONVENTIONS

- Experiments & Diagnostics - CESM1.2 experiment results and diagnostices will be added to the site as they become available.
- Post Processing Utilities
- Model File Naming Conventions
- Experiment Case Naming Conventions

MODEL DOCUMENTATION

EXTERNAL LIBRARY DOCUMENTATION

- Parallel I/O Library (PIO)
- Model Coupling Toolkit (MCT)
- Earth System Modeling Framework (ESMF)

MODEL INPUT DATA

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

PERFORMANCE AND LOAD BALANCING DATA

The timing table provides performance data that will continue to evolve due to changes in the model, machine hardware and input from the user community. For CESM1.2, please refer to the CESM1.1.1 Timing Table.

CESM PROJECT

The Community Earth System Model (CESM) is a fully-coupled, global climate model that provides state-of-the-art computer simulations of the Earth's past, present, and future climate states.

CESM is sponsored by the National Science Foundation (NSF) and the U.S. Department of Energy (DOE). Administration of the CESM is maintained by the Climate and Global Dynamics Division (CGD) at the National Center for Atmospheric Research (NCAR).

Background and Sponsors

MODEL SOURCE CODE

Copyright and Terms of Use

All CESM source code is subject to the following Copyright Notice and Disclaimer.

CESM Release Versions

CESM X.Y.Z - CESM model release versions include three numbers separated by a dot (.) where

- X - corresponds to the major release number indicating significant science changes.
- Y - corresponds to the addition of new infrastructure and new science capabilities for targeted components.
- Z - corresponds to release bug fixes and machine updates.

Release versions

Acquiring the Release Code

The source code for CESM releases is distributed through a public Subversion code repository. This code can be checked out using Subversion client software, such as the command tool svn, or simply view the latest version with a web browser.

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

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

How to Acquire the Code

REPORTING A PROBLEM

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

Reporting Problems, Known Problems

CESM SUPPORT POLICY

CESM Support Policy - November 2012

CESM DATA MANAGEMENT & DISTRIBUTION PLAN

The Community Earth System Model (CESM) Data Management and Data Distribution Plan documents the procedures for the storage and distribution of data associated with the CESM project.

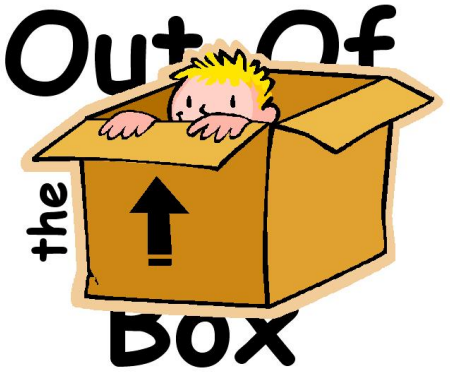
CESM data management and distribution

Software/Hardware Requirements

- **Subversion client (version 1.4.2 or greater)**
- **Fortran and C compilers (recommend PGI, Intel, or IBM's XLF)**
- **NetCDF library (recommend version 4.1.3 or later)**
- **MPI (MPI1 is adequate; OpenMPI or MPICH seem to work on Linux clusters)**

[Note: Other external libraries (ESMF, MCT, PIO) are included in CESM source code, and do not have to be separately installed.]

- **CESM currently runs "out of the box" today on the following machines**
  - **yellowstone – NCAR IBM**
  - **titan – ORNL Cray XK6**
  - **hopper – NERSC Cray XE6**
  - **edison – NERSC Cray Cascade**
  - **bluewaters – ORNL Cray XE6**
  - **intrepid – ANL IBM Bluegene/P**
  - **mira – ANL IBM Bluegene/Q**
  - **janus – Univ Colorado HPC cluster**
  - **pleiades – NASA SGI ICE cluster**
  - **and a few others**



Basic Work Flow

(*or how to set up and run an experiment)*

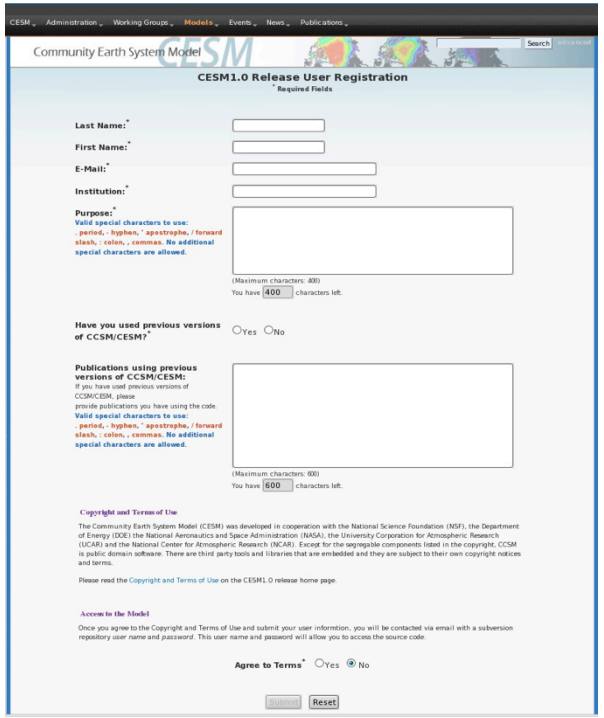
- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: Initial Run and Output Data Flow**
  - **(5) Run the Model: Continuation Run(s)**

(A) Registration

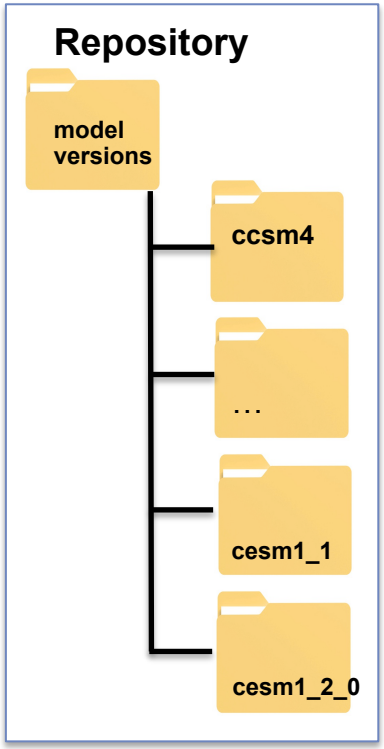
- **Go to CESM1.2 home page: http://www.cesm.ucar.edu/models/cesm1.2/**

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



- **Register -- you will be emailed a username and password**

(A) Download the Source Code



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

• **List the versions available on the CESM repository**

```
svn list https://svn-ccsm-release.cgd.ucar.edu/model_versions
```

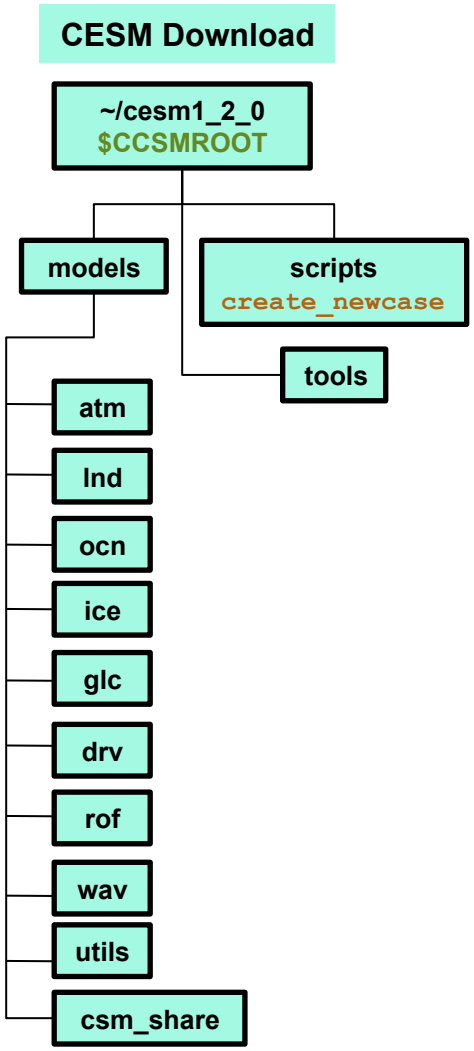
• **Check out a working copy from the repository (“Download code”)**

```
svn co https://svn-ccsm-release.cgd.ucar.edu/model_versions/cesm1_2_0
```

(*) You can get subversion at http://subversion.apache.org/

open source

(A) Overview of Directories (after initial model download)



Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: 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 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

- 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

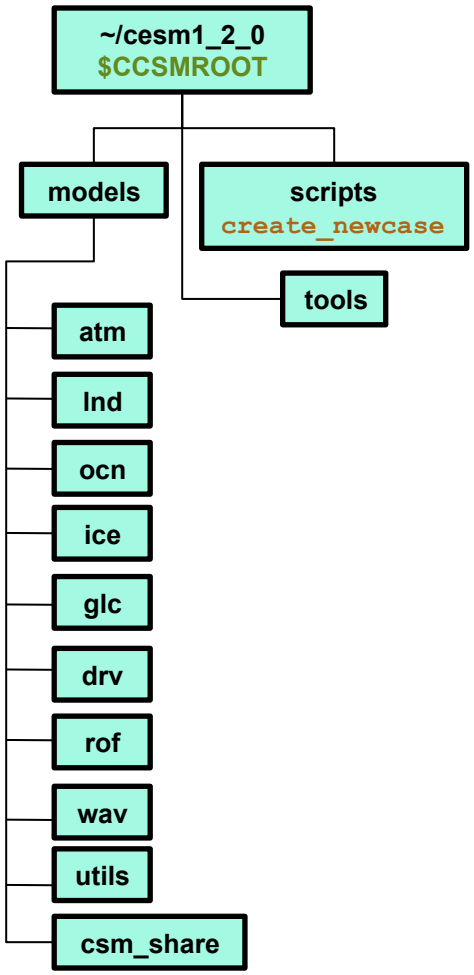
- *Do NOT attempt to download or check out this entire repository by hand!*

(B) Overview of Directories (+ inputdata directory)

INPUTDATA Directory

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

CESM Download



Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: 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

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine: “yellowstone”

```
# go to root directory of source code download
cd /glade/p/cesm/tutorial/cesm1_2_0.tutorial

# go into scripts subdirectory
cd scripts

# (1) make a subdirectory  for your cases in your home directory and create a new case
 mkdir ~/cases  (this only needs to be done once)
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B_1850_CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

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

# (4) submit an initial (startup) run to the batch queue
./b.day1.0.submit
```

Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: Initial Run and Output Data Flow**
  - **(5) Run the Model: Continuation Run(s)**

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine: “yellowstone”

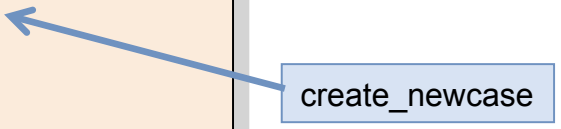
```
# go to root directory of source code download
cd /glade/p/cesm/tutorial/cesm1_2_0.tutorial

# go into scripts subdirectory
cd scripts

# (1) make a subdirectory for your cases in your home directory and create a new case
 mkdir ~/cases
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B_1850_CN -mach yellowstone


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

# (2) invoke cesm_setup
./cesm_setup

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

# (4) submit an initial run to the batch queue
./cases01.submit
```

(1) Create a New Case

- **Go to the scripts directory: …/cesm1_2_0/scripts/**
- **Scripts are a combination of Perl, csh, and sh, with XML configuration files.**

- **create_newcase is the tool that generates a new case**

```
cd …/cesm1_2_0/scripts/

drwxr-sr-x 5 jshollen cseg 131072 May  7 13:53 .
drwxr-sr-x 6 jshollen cseg 131072 May  7 13:53 ..
drwxr-sr-x 8 jshollen cseg 131072 May  7 13:53 ccsm_utils
-rw-r--r-- 1 jshollen cseg 581940 May  7 13:53 ChangeLog
-rwxr-xr-x 1 jshollen cseg  19229 May  7 13:53 create_clone
-rwxr-xr-x 1 jshollen cseg  81134 May  7 13:53 create_newcase
-rwxr-xr-x 1 jshollen cseg  54590 May  7 13:53 create_test
drwxr-sr-x 5 jshollen cseg 131072 May  7 13:53 doc
-rwxr-xr-x 1 jshollen cseg   1255 May  7 13:53 link_dirtree
-rwxr-xr-x 1 jshollen cseg  12701 May  7 13:53 query_tests
-rw-r--r-- 1 jshollen cseg   2345 May  7 13:53 README
-rw-r--r-- 1 jshollen cseg   1113 May  7 13:53 sample_pes_file.xml
drwxr-sr-x 6 jshollen cseg 131072 May  7 13:53 .svn
-rw-r--r-- 1 jshollen cseg    203 May  7 13:53 SVN_EXTERNAL_DIRECTORIES
```



(1) About create_newcase

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

```
cd …/cesm1_2_0/scripts/
./create_newcase –help

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

     -case <name>          Specifies the case name (required).
     -compset <name>       Specify a CESM compset (required).
     -res <name>           Specify a CESM grid resolution (required).
     -mach <name>          Specify a CESM machine (required).
     -compiler <name>      Specify a compiler for the target machine (optional)
                           default: default compiler for the target machine
     -mpilib <name>        Specify a mpi library for the target machine (optional)
                           default: default mpi library for the target machine
                           allowed: openmpi, mpich, ibm, mpi-serial, etc
                           redundant with _M confopts setting
     -mach_dir <path>      Specify the locations of the Machines directory (optional).
                           default: /glade/p/cesm/cseg/collections/cesm1_2_0_beta08/scripts/ccsm_utils/Machines

     -pecount <name>       Value of S,M,L,X1,X2 (optional).
                           default: M, partially redundant with confopts _P
     -pes_file <name>      Full pathname of pes file to use (will overwrite default settings) (optional).
                           See sample_pes_file.xml for an example.
     -user_compset         Long name for new user compset file to use (optional)
                           This assumes that all of the compset settings in the long name have been defined.
     -grid_file <name>     Full pathname of grid file to use (optional)
                           See sample_grid_file.xml for an example.
                           Note that compset components must support the new grid.
     -help [or -h]         Print usage to STDOUT (optional).
     -list <type>          Only list valid values, type can be [compsets, grids, machines] (optional).

     ...
```

(1) About create_newcase

The command create_newcase has 4 required arguments.

```
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 \
        -compset B_1850_CN -mach yellowstone
```

- **"case" is the name and location of the case being created**
  - **~/cases/b.day1.0**

- **"res" specifies the model resolutions (or grid)**
  - **Each model resolution can be specified by its alias, short name and long name.**
  - **Example of equivalent alias, short name and long name:**
  - **alias: T31_g37**
  - **short name: T31_gx3v7**
  - **long name = a%T31_l%T31_oi%gx3v7_r%r05_m%gx3v7_g%null_w%null**

(1) About create_newcase

- **"compset" specifies the "component set"**
  - **component set specifies component models, forcing scenarios and physics options for those models**
  - **Each model compset can be specified by its alias, short name and long name.**

  Example of equivalent alias, short name and long name:
  - **alias: B1850CN**
  - **short name: B_1850_CN**
  - **long name = 1850_CAM4_CLM40%CN_CICE_POP2_RTM_SGLC_SWAV**

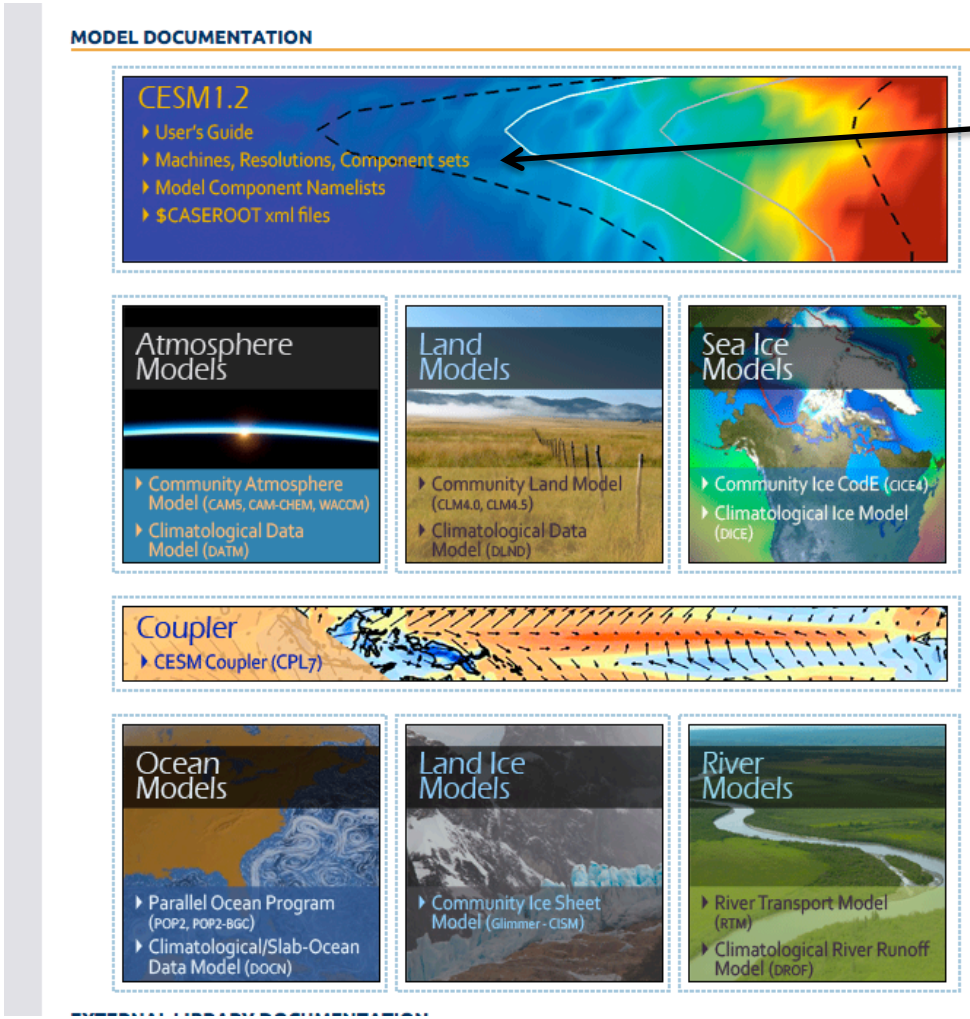
- **"mach" specifies the machine that will be used.**
  - **"supported" machines tested regularly, eg. yellowstone, titan, hopper, intrepid**
  - **"generic machines" provide a starting point for porting, eg. generic_ibm**

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

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

`./create_newcase -list <type>`

with type can be [compsets, grids, machines]



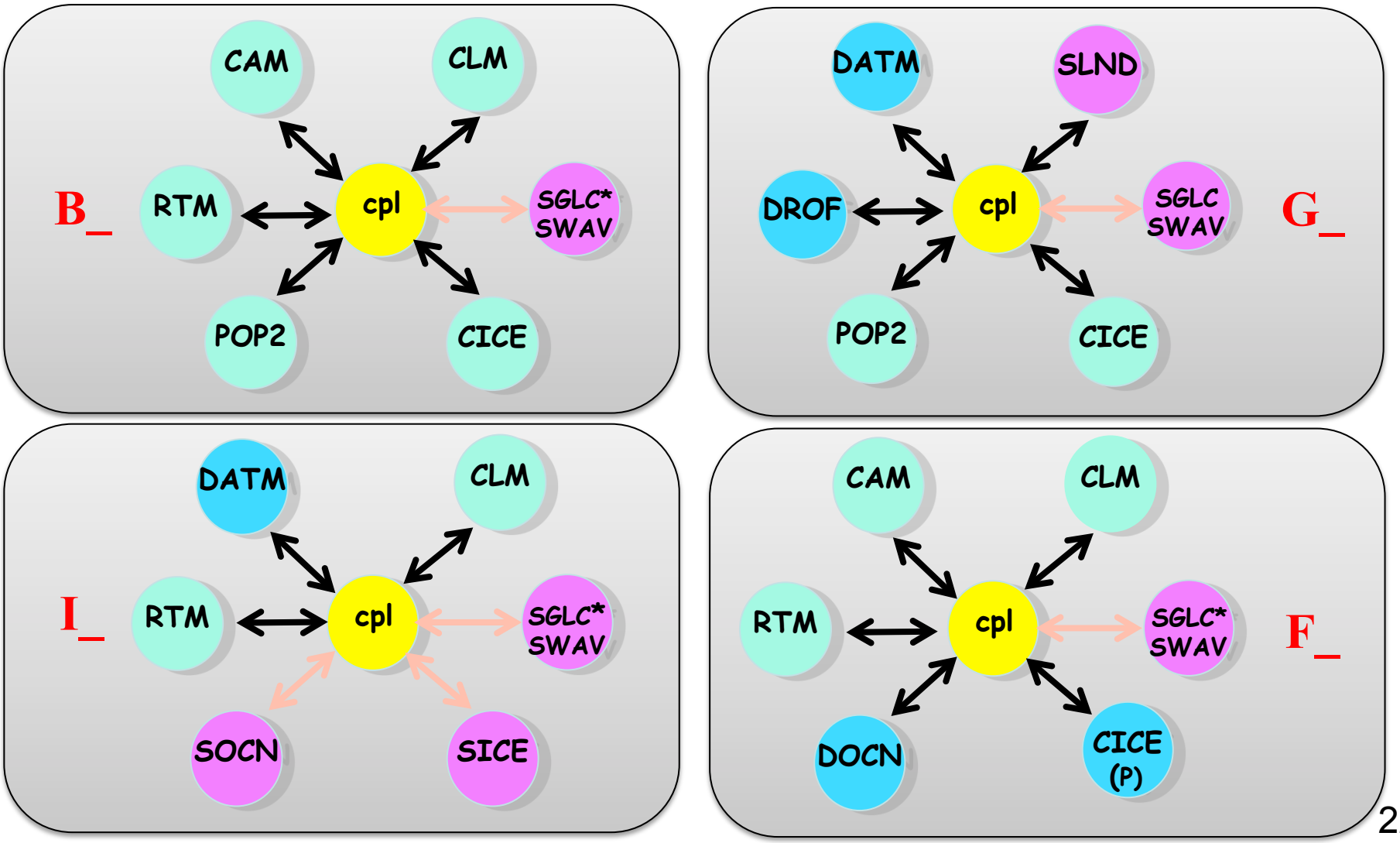
List of valid values is also available from the CESM website

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

More on CESM component sets

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

***some of the B, I, E, and F compsets include the full CISM1 (land ice) model**



(1) Result of running create_newcase

```
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 \
-compset B_1850_CN -mach yellowstone
```

```
--------------------------------------------------------------------------------
For a list of potential issues in the current tag, please point your web browser to:
https://svn-ccsm-models.cgd.ucar.edu/cesm1/known_problems/
--------------------------------------------------------------------------------
 grid longname is T31_g37
Component set: longname (shortname) (alias)
  1850_CAM4_CLM40%CN_CICE_POP2_RTM_SGLC_SWAV (B_1850_CN) (B1850CN)
Component set Description:
  CAM: CLM: RTM: CICE: POP2: SGLC: SWAV: pre-industrial: cam4 physics: clm4.0
physics: clm4.0 cn specified phenology: prognostic cice: POP2 default:
Grid:
  a%T31_l%T31_oi%gx3v7_r%r05_m%gx3v7_g%null_w%null (T31_gx3v7)
  ATM_GRID = 48x96 NX_ATM=96 NY_ATM=48
  LND_GRID = 48x96 NX_LND=96 NX_LND=48
  …

Non-Default Options:
  ATM_NCPL: 48
  BUDGETS: TRUE
  CAM_CONFIG_OPTS: -phys cam4
  …
The PE layout for this case match these options:
GRID =  a%T31.+oi%gx3
CCSM_LCOMPSET =  CAM.+CLM.+CICE.+POP
MACH =  yellowstone
Creating /glade/u/home/hannay/cases/b.day1.0
Created /glade/u/home/hannay/cases/b.day1.0/env_case.xml
Created /glade/u/home/hannay/cases/b.day1.0/env_mach_pes.xml
Created /glade/u/home/hannay/cases/b.day1.0/env_build.xml
Created /glade/u/home/hannay/cases/b.day1.0/env_run.xml
Locking file /glade/u/home/hannay/cases/b.day1.0/env_case.xml
Successfully created the case for yellowstone

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

compset info

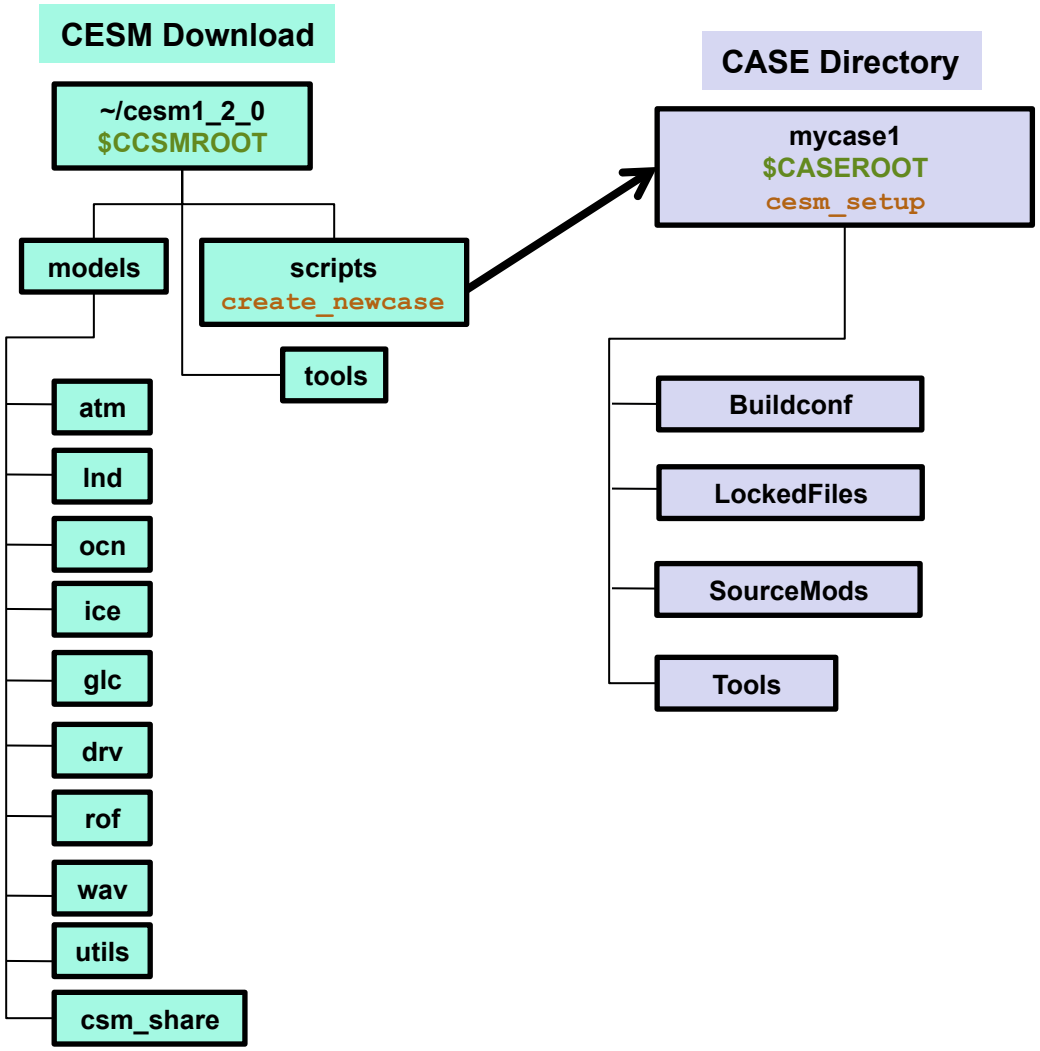
grid info

non default options

case location

Success!

(1) Overview of Directories (after create_newcase)



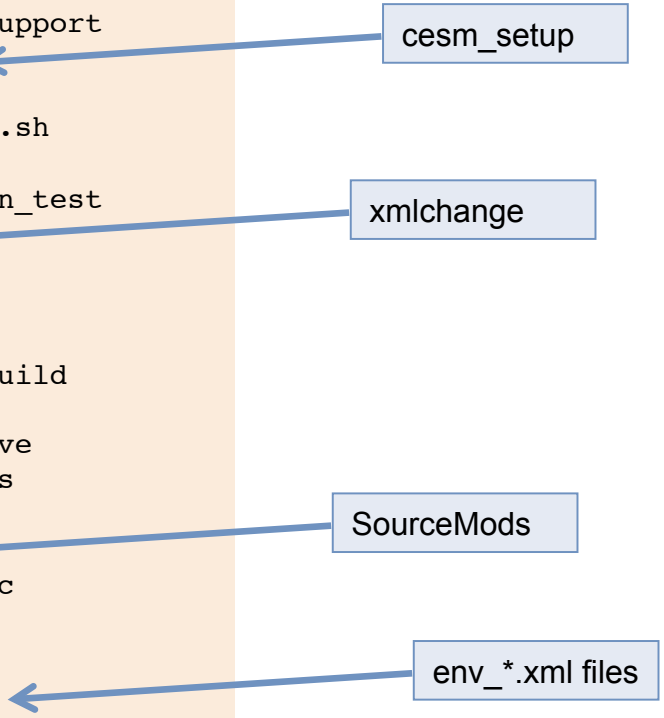
INPUTDATA Directory

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

(1) Case directory after running
create_newcase

- **SourceMods - directory for case specific code modifications**
- **env_*.xml - contains xml/environment variables**
- **xmlchange - script that changes xml (env) variable values**
- **cesm_setup - script used in the next step, step (2)**

```
-rw-rw-r--  1 hannay ncar  1500 Jun 10 09:03 README.case
-rw-r--r--  1 hannay ncar  2345 Jun 10 09:03 README.science_support
-rwxr-xr-x  1 hannay cseg 14495 Jun 10 09:03 cesm_setup
-rwxr-xr-x  1 hannay cseg 10126 Jun 10 09:03 check_input_data
-rwxr-xr-x  1 hannay cseg 15390 Jun 10 09:03 archive_metadata.sh
-rwxr-xr-x  1 hannay cseg   837 Jun 10 09:03 check_case
-rwxr-xr-x  1 hannay cseg  3672 Jun 10 09:03 create_production_test
-rwxr-xr-x  1 hannay cseg 12569 Jun 10 09:03 xmlchange
-rwxr-xr-x  1 hannay cseg 10503 Jun 10 09:03 xmlquery
drwxrwxr-x  3 hannay ncar 16384 Jun 10 09:03 Tools
-rwxr-xr-x  1 hannay ncar 13233 Jun 10 09:03 b.day1.0.build
-rwxr-xr-x  1 hannay ncar  1048 Jun 10 09:03 b.day1.0.clean_build
-rwxr-xr-x  1 hannay ncar   608 Jun 10 09:03 b.day1.0.submit
-rwxrwxr-x  1 hannay ncar   918 Jun 10 09:03 b.day1.0.l_archive
-rwxr-xr-x  1 hannay ncar  2127 Jun 10 09:03 preview_namelists
drwxrwxr-x  2 hannay ncar 16384 Jun 10 09:03 Buildconf
drwxrwxr-x 11 hannay ncar 16384 Jun 10 09:03 SourceMods
-rwxr-xr-x  1 hannay ncar  2653 Jun 10 09:03 env_mach_specific
-rw-r--r--  1 hannay ncar   301 Jun 10 09:03 Depends.intel
-rw-rw-r--  1 hannay ncar  4421 Jun 10 09:03 env_case.xml
-rw-rw-r--  1 hannay ncar  6998 Jun 10 09:03 env_mach_pes.xml
-rw-rw-r--  1 hannay ncar 10849 Jun 10 09:03 env_build.xml
-rw-rw-r--  1 hannay ncar 23197 Jun 10 09:03 env_run.xml
drwxrwxr-x  2 hannay ncar 16384 Jun 10 09:03 LockedFiles
-rw-rw-r--  1 hannay ncar   135 Jun 10 09:03 CaseStatus
```



About .xml Files: Format & Variables

- Contains variables used by scripts -- some can be changed by the user
- Here's a snippet of the env_run.xml file

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

<!--"sets the run length in conjunction with STOP_OPTION and STOP_DATE (integer) " -->
<entry id="STOP_N"   value="5"  />

<!--"logical to turn on short term archiving, valid values: TRUE,FALSE (logical) " -->
<entry id="DOUT_S"   value="TRUE"  />

<!--"local short term archiving root directory (char) " -->
<entry id="DOUT_S_ROOT"   value="/ptmp/$CCSMUSER/archive/$CASE"  />
```

- "id" - variable name
- "value" – variable value
- <!--- text --> description above the entry
- To modify a variable in an xml file – use xmlchange

```
> xmlchange -help
> xmlchange STOP_N=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_mach_pes.xml
  - Specifies layout of components on hardware processors
  - Use this to tune performance - scientific results do not depend on component/ processor layout
- env_build.xml
  - Specifies build information including component resolutions and component configuration options
- env_run.xml
  - Sets run time information (such as length of run, frequency of restarts, output of coupler diagnostics, and short-term and long-term archiving.)
  - User interacts with this file most frequently
- env_mach_specific
  - Actually a shell script rather than an XML file.
  - Sets modules and paths to libraries (e.g. MPI)
  - Can change compiler options, libraries, etc.
  - Part of porting is to set variables here

Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: Initial Run and Output Data Flow**
  - **(5) Run the Model: Continuation Run(s)**

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine: “yellowstone”

```
# go to root directory of source code download
cd /glade/p/cesm/tutorial/cesm1_2_0.tutorial

# go into scripts subdirectory
cd scripts

# (1) make a subdirectory for your cases in your home directory and create a new case
 mkdir ~/cases
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B_1850_CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

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

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

(2) The command “cesm_setup”

The command: cesm_setup

- **Creates the Macros file if it does not exist. This specifies Makefile compilation variables (e.g. FFLAGS).**

- **Creates the namelist modification files user_nl_xxx, (where xxx denotes the set of components targeted for the specific case)**

- **Creates a batch script for on the number of processors required by env_mach_pes.xml settings: *.run**

- **Creates the directory CaseDocs:**
  - **Contains all the a documentation copy of the component namelists.**
  - ***This is for reference only and files in this directory SHOULD NOT BE EDITED.***

(2) About cesm_setup

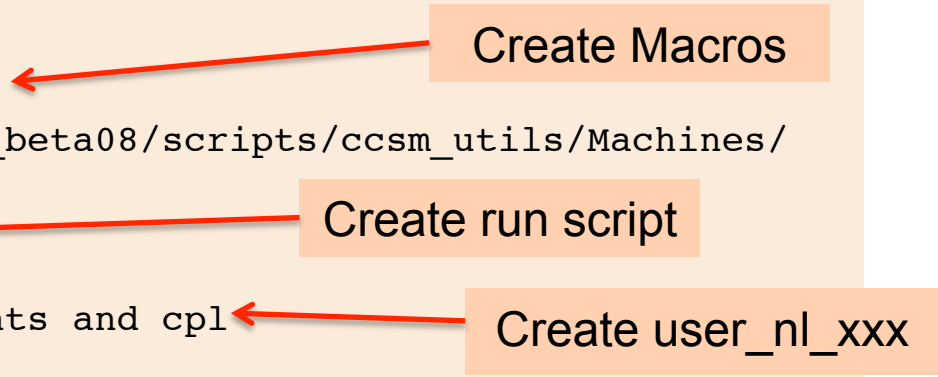
```
./cesm_setup -help
```

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

USAGE
     cesm_setup [options]

OPTIONS
     -help [or -h]         Print usage to STDOUT.

     -clean                Removes the batch run script for target machines
                           Macros and user_nl_xxx files are never removed
                           by cesm_setup - you must remove them manually
```

(2) Calling cesm_setup

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

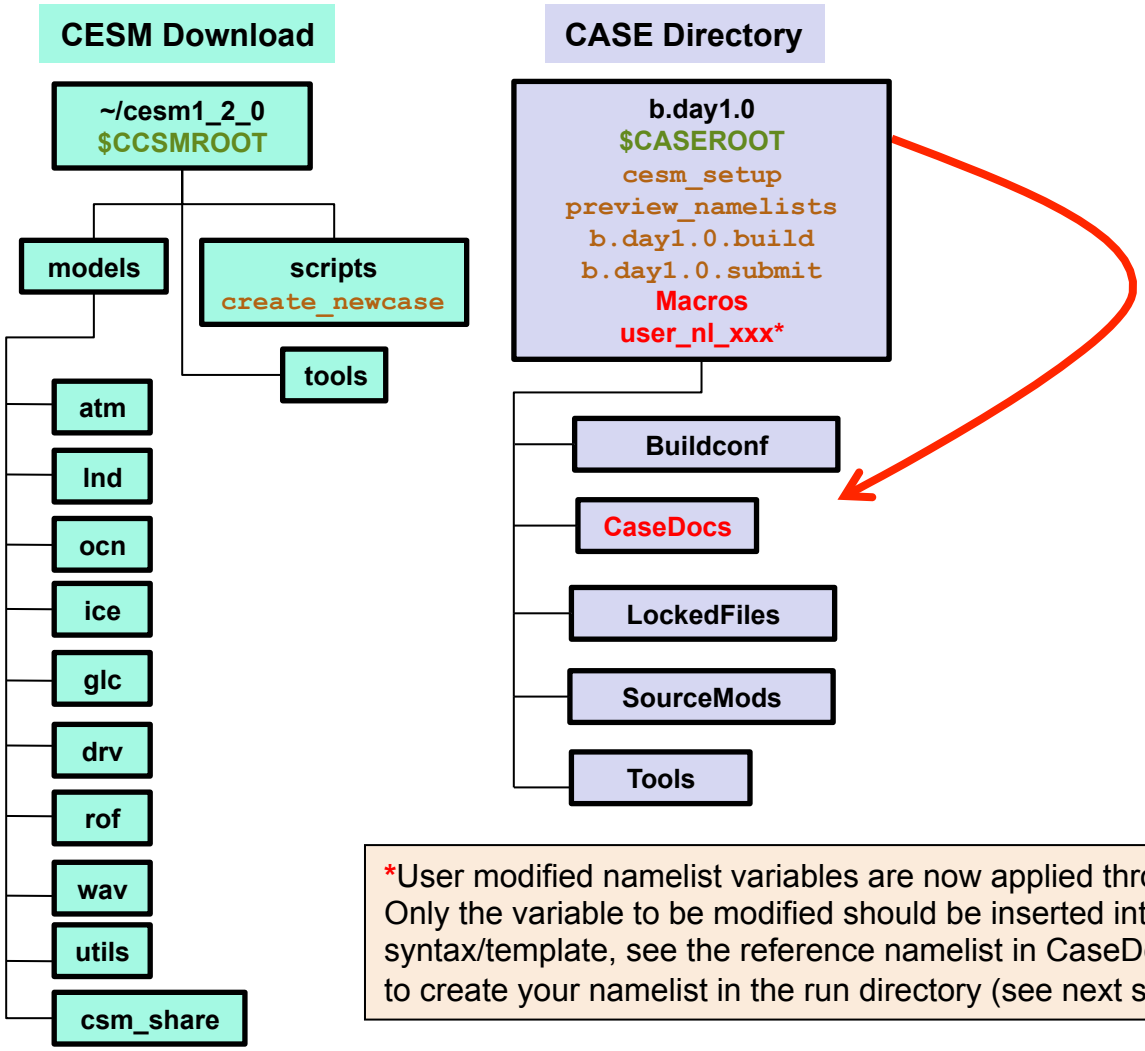
```
Creating Macros file for yellowstone
/glade/p/cesm/cseg/collections/cesm1_2_beta08/scripts/ccsm_utils/Machines/
config_compilers.xml intel yellowstone
Creating batch script b.day1.0.run
Locking file env_mach_pes.xml
Creating user_nl_xxx files for components and cpl
Running preview_namelist script
 infile is /glade/u/home/hannay/cases/b.day1.0/Buildconf/cplconf/
cesm_namelist
CAM writing dry deposition namelist to drv_flds_in
CAM writing namelist to atm_in
CLM configure done.
CLM adding use_case 1850_control defaults for var sim_year with val 1850
CLM adding use_case 1850_control defaults for var sim_year_range with val
constant
CLM adding use_case 1850_control defaults for var use_case_desc with val
Conditions to simulate 1850 land-use
CICE configure done.
POP2 build-namelist: ocn_grid is gx1v6
POP2 build-namelist: ocn_tracer_modules are  iage
See ./CaseDoc for component namelists
If an old case build already exists, might want to run b.day1.0.clean_build
before building
```



(2) Overview of Directories (after cesm_setup)

INPUTDATA Directory

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



*User modified namelist variables are now applied through the user_nl_xxx files. Only the variable to be modified should be inserted into this file. To find the proper syntax/template, see the reference namelist in CaseDocs, or use "preview_namelist" to create your namelist in the run directory (see next sections).

Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: Initial Run and Output Data Flow**
  - **(5) Run the Model: Continuation Run(s)**

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine: “yellowstone”

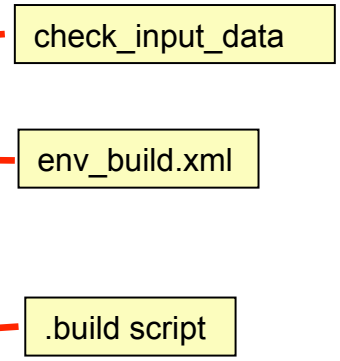
```
# go to root directory of source code download
cd /glade/p/cesm/tutorial/cesm1_2_0.tutorial

# go into scripts subdirectory
cd scripts

# (1) make a subdirectory for your cases in your home directory and create a new case
 mkdir ~/cases
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B_1850_CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

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

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

(3) Build the Model

- Use the *.build script
- Modifications before build
  - Change env_build.xml values *before* running *.build
  - Introduce modified source code in SourceMods/ before building
- To completely rebuild, run *.clean_build first
- The *.build script
  - Checks for missing input data
  - Creates directory for executable code and model namelist files
  - Locks env_build.xml
  - Builds the individual component libraries and model executable
- If any inputdata is missing,
  - Build aborts, but provides a list of missing files
  - Run ./check_input_data –export to acquire missing data
  - This will use svn to put required data in the inputdata directory
  - Then re-run build script

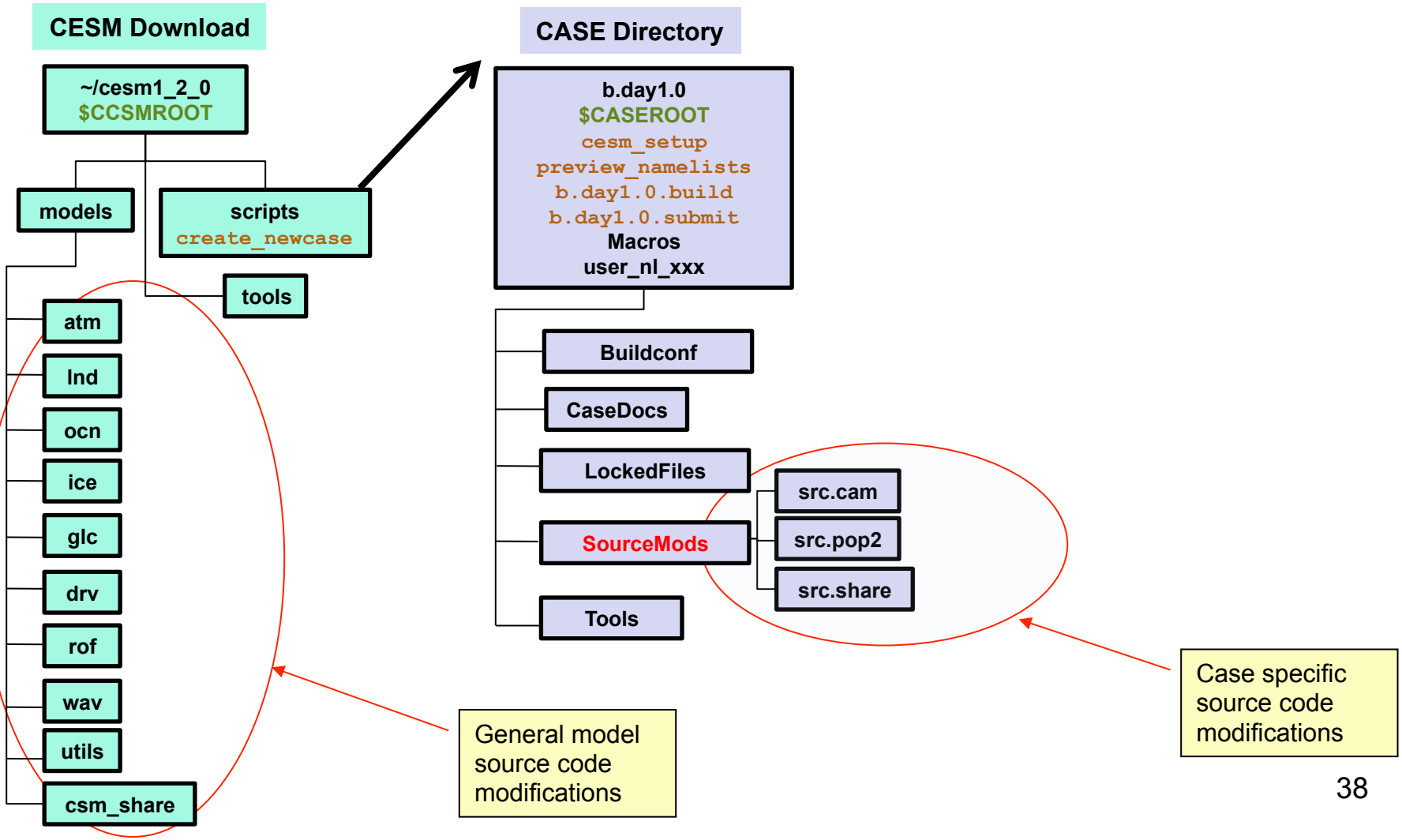
(3) The *.build script

```
cases/b.day1.0>ls -l
total 432

<snippet>
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
drwxr-xr-x    2 userx   ncar          8192 May 13 14:32 README.scence_support
-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 cesm_setup
-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           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 b.day1.0.build
-rwxrwxr-x    1 userx   ncar           836 May 13 17:12 b.day1.0.clean_build
-rwxrwxr-x    1 userx   ncar           802 May 13 17:12 b.day1.0.l_archive
-rwxrwxr-x    1 userx   ncar          3938 May 13 17:12 b.day1.0.run
-rwxrwxr-x    1 userx   ncar           608 May 13 17:12 b.day1.0.submit
-rwxr-xr-x    1 userx   ncar         10388 May 12 11:33 xmlchange
<snippet>
```



(3) Modifying Source Code

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



(3) Running the .build Script

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

```
./b.day1.0.build
-------------------------------------------------------------------
 CESM BUILDNML SCRIPT STARTING
 - To prestage restarts, untar a restart.tar file into /glade/scratch/hannay/b.day1.0/run
 infile is /glade/u/home/hannay/cases/b.day1.0/Buildconf/cplconf/cesm_namelist
. . .
 CESM BUILDNML SCRIPT HAS FINISHED SUCCESSFULLY
-------------------------------------------------------------------
-------------------------------------------------------------------
 CESM PRESTAGE SCRIPT STARTING
 - Case input data directory, DIN_LOC_ROOT, is /glade/p/cesm/cseg//inputdata
 - Checking the existence of input datasets in DIN_LOC_ROOT
 CESM PRESTAGE SCRIPT HAS FINISHED SUCCESSFULLY
-------------------------------------------------------------------
-------------------------------------------------------------------
 CESM BUILDEXE SCRIPT STARTING
 COMPILER is intel
 - Build Libraries: mct gptl pio csm_share
Tue Jun 11 19:13:41 MDT 2013 /glade/scratch/hannay/b.day1.0/bld/mct/mct.bldlog.130611-191330
. . .
 - Locking file env_build.xml
 CESM BUILDEXE SCRIPT HAS FINISHED SUCCESSFULLY
-------------------------------------------------------------------
```

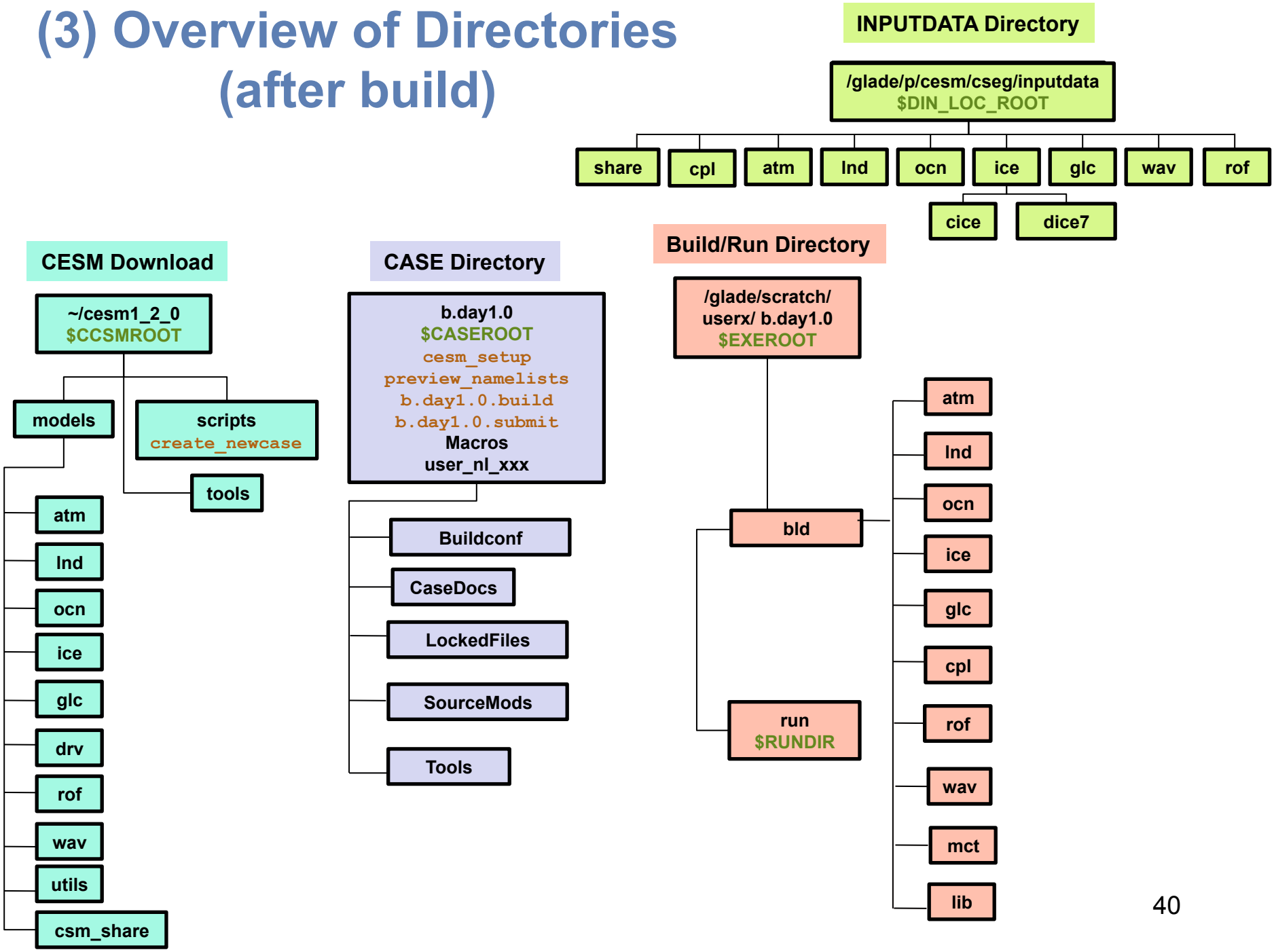
Namelist creation

Inputdata verification and prestage

Model Build

Success

(3) Overview of Directories (after build)



Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: Initial Run and Output Data Flow**
  - **(5) Run the Model: Continuation Run(s)**

Work Flow: Super Quick Start

These unix commands built and ran the model on a supported machine: “yellowstone”

```
# go to root directory of source code download
cd /glade/p/cesm/tutorial/cesm1_2_0.tutorial

# go into scripts subdirectory
cd scripts

# (1) make a subdirectory for your cases in your home directory and create a new case
 mkdir ~/cases
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B_1850_CN -mach yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

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

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

(4) Running the Model: Initial Run

- May want to edit `env_run.xml` file before running (e.g. change run length)
- May also want to modify component namelist settings
- Can change `env_run.xml` variables
- Or modify a namelist through user_nl_xxx
- 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/b.day1.0> b.day1.0.submit
```

```
cases/b.day1.0>b.day1.0.submit
check_case OK
Job <40597> is submitted to queue <regular>.
```

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

```
cases/mycase1>bjobs
JOBID   USER    STAT  QUEUE     FROM_HOST      EXEC_HOST   JOB_NAME   SUBMIT_TIME
40597   userx   PEND  regular    yslogin1-ib   15*ys1358   b.day1.0     Jun 12
18:30
                                                           .
                                                           .
                                                           .
```

(4) Output in Your CASE Directory

```
~/cases/b.day1.0 > ls –l
-rwxr-xr-x  1 hannay cseg 15390 May  7 13:53 archive_metadata.sh
drwxrwxr-x  8 hannay ncar 16384 Jun 12 21:29 Buildconf
-rwxr-xr-x  1 hannay ncar 13233 Jun 10 21:38 b.day1.0.build
-rwxr-xr-x  1 hannay ncar  1048 Jun 10 21:38 b.day1.0.clean_build
-rwxrwxr-x  1 hannay ncar   918 Jun 10 21:38 b.day1.0.l_archive
-rwxr-xr-x  1 hannay ncar 10270 Jun 12 21:28 b.day1.0.run
-rwxr-xr-x  1 hannay ncar   608 Jun 10 21:38 b.day1.0.submit
drwxrwxr-x  2 hannay ncar 16384 Jun 10 21:38 CaseDocs
-rw-rw-r--  1 hannay ncar   270 Jun 12 21:29 CaseStatus
-rwxr-xr-x  1 hannay cseg 14495 May  7 13:53 cesm_setup
-rw-rw-r--  1 hannay ncar     0 Jun 12 21:29 cesm.stderr.920879
-rw-rw-r--  1 hannay ncar  1300 Jun 12 21:29 cesm.stdout.920879
-rwxr-xr-x  1 hannay cseg   837 May  7 13:53 check_case
-rwxr-xr-x  1 hannay cseg 10126 May  7 13:53 check_input_data
-rw-rw-r--  1 hannay ncar 10924 Jun 12 21:29 env_build.xml
-rw-rw-r--  1 hannay ncar  4421 Jun 10 21:38 env_case.xml
-rw-rw-r--  1 hannay ncar   895 Jun 12 21:29 env_derived
-rw-rw-r--  1 hannay ncar  7003 Jun 10 21:38 env_mach_pes.xml
-rwxr-xr-x  1 hannay ncar  2653 Jun 10 21:38 env_mach_specific
-rw-rw-r--  1 hannay ncar 23197 Jun 10 21:38 env_run.xml
-rw-rw-r--  1 hannay ncar     9 Jun 12 21:29 hostfile
drwxrwxr-x  2 hannay ncar 16384 Jun 11 19:23 LockedFiles
drwxrwxr-x  3 hannay ncar 16384 Jun 11 19:23 logs
-rw-rw-r--  1 hannay ncar   954 Jun 10 21:38 Macros
-rwxr-xr-x  1 hannay ncar  2127 Jun 10 21:38 preview_namelists
-rw-rw-r--  1 hannay ncar  1500 Jun 10 21:37 README.case
-rw-r--r--  1 hannay ncar  2345 Jun 10 21:37 README.science_support
drwxrwxr-x 11 hannay ncar 16384 Jun 10 21:38 SourceMods
drwxrwxr-x  2 hannay ncar 16384 Jun 12 21:31 timing
drwxrwxr-x  3 hannay ncar 16384 Jun 10 21:38 Tools
-rw-r--r--  1 hannay ncar   115 Jun 10 21:38 user_nl_cam
-rw-r--r--  1 hannay ncar   367 Jun 10 21:38 user_nl_cice
-rw-r--r--  1 hannay ncar  1040 Jun 10 21:38 user_nl_clm
-rw-r--r--  1 hannay ncar  2284 Jun 10 21:38 user_nl_cpl
-rw-r--r--  1 hannay ncar  2949 Jun 10 21:38 user_nl_pop2
-rw-r--r--  1 hannay ncar   573 Jun 10 21:38 user_nl_rtm
-rwxr-xr-x  1 hannay cseg 12569 May  7 13:53 xmlchange
-rwxr-xr-x  1 hannay cseg 10503 May  7 13:53 xmlquery
```

stdout/err

Log files

Timing files

(4) Output in Your CASE Directory

```
~/cases/b.day1.0/logs > ls -l
-rw-rw-r-- 1 hannay ncar 37047 Jun 12 21:31 atm.log.130612-212912.gz
drwxrwxr-x 2 hannay ncar 16384 Jun 11 19:23 bld
-rw-rw-r-- 1 hannay ncar 24235 Jun 12 21:31 cesm.log.130612-212912.gz
-rw-rw-r-- 1 hannay ncar  6696 Jun 12 21:31 cpl.log.130612-212912.gz
-rw-rw-r-- 1 hannay ncar 17074 Jun 12 21:31 ice.log.130612-212912.gz
-rw-rw-r-- 1 hannay ncar  7810 Jun 12 21:31 lnd.log.130612-212912.gz
-rw-rw-r-- 1 hannay ncar 20175 Jun 12 21:31 ocn.log.130612-212912.gz
-rw-rw-r-- 1 hannay ncar  1772 Jun 12 21:31 rof.log.130612-212912.gz
```

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

```
~/cases/b.day1.0/timing > ls -l
-rw-rw-r-- 1 hannay ncar 7898 Jun 12 21:31
ccsm_timing.b.day1.0.130612-212912
-rw-rw-r-- 1 hannay ncar 9844 Jun 12 21:31 ccsm_timing_stats.
130612-212912.gz
```

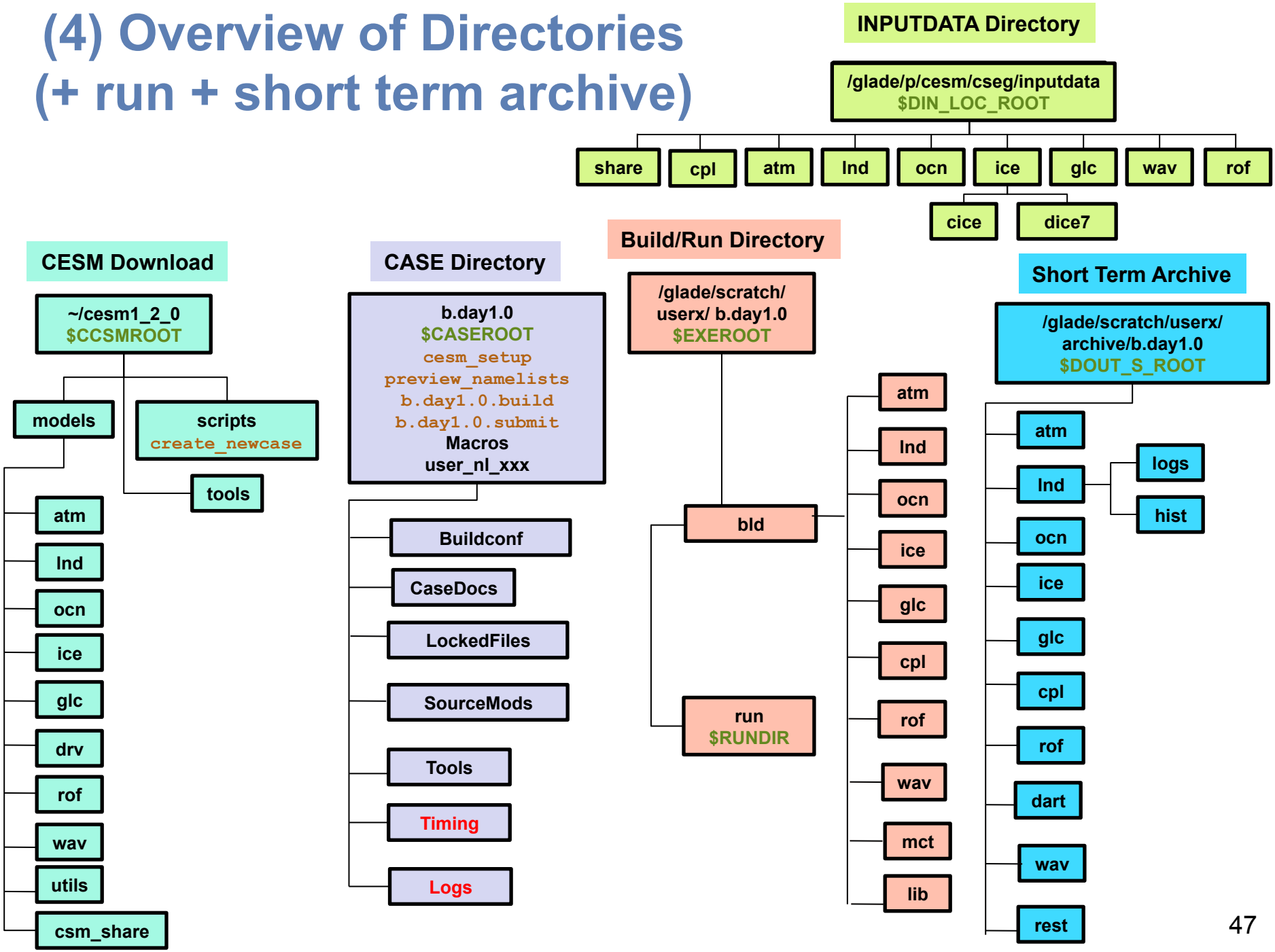
Timing files tells about model throughput (how many model years per day) and model cost (pe-hrs per simulated years).
Each time a job is run a new timing file is created in this directory.

(4) Output in Short Term Archiving Directory

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

```
cases/b.day1.0>echo $DOUT_S_ROOT
/glade/scratch/userx/archive/b.day1.0
cases/b.day1.0>ls -l $DOUT_S_ROOT
total 3072
drwxr-xr-x 12 shields ncar 131072 Jun 12 18:08 .
drwxr-xr-x  7 shields ncar 131072 Jun 12 15:04 ..
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 atm
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 cpl
drwxr-xr-x  5 shields ncar 131072 Jun 12 18:08 dart
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 glc
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 ice
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 lnd
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 ocn
drwxr-xr-x  3 shields ncar 131072 Jun 12 18:08 rest
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 rof
drwxr-xr-x  4 shields ncar 131072 Jun 12 18:08 wav
cases/b.day1.0>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/b.day1.0>ls -l $DOUT_S_ROOT/cpl/logs/
total 256
-rw-r--r--    1 userx   ncar          19115 May 18 18:37 cesm.log.100518-183212.gz
-rw-r--r--    1 userx   ncar           4998 May 18 18:37 cpl.log.100518-183212.gz
cases/b.day1.0>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)



Basic Work Flow

(*or how to set up and run an experiment)*

- **One-Time Setup Steps**
  - **(A) Registration and Download**
  - **(B) Create an Input Data Root Directory**
  - **(C) Porting**

- **Creating & Running a Case**
  - **(1) Create a New Case**
  - **(2) Invoke cesm_setup**
  - **(3) Build the Executable**
  - **(4) Run the Model: Initial Run and Output Data Flow**
  - **(5) Run the Model: Continuation Run(s)**

Work Flow: Super Quick Start

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

```
# go to root directory of source code download
cd /glade/p/cesm/tutorial/cesm1_2_0.tutorial/

# go into scripts subdir
cd scripts

# (1)make a subdirectory for your cases in your home dir and create a new case
mkdir ~/cases
./create_newcase -case ~/cases/b.day1.0 -res T31_g37 -compset B_1850_CN -mach
yellowstone

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

# (2) invoke cesm_setup
./cesm_setup

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

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

# 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 CONTINUE_RUN=TRUE

# (5) submit a continuation run to the batch queue
./b.day1.0.submit

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

(5) Running the Model: Continuation Runs

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

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

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

- **May want to turn on "long term archiving" in env_run.xml (DOUT_L_MS)**

(5) Long Term Archiving

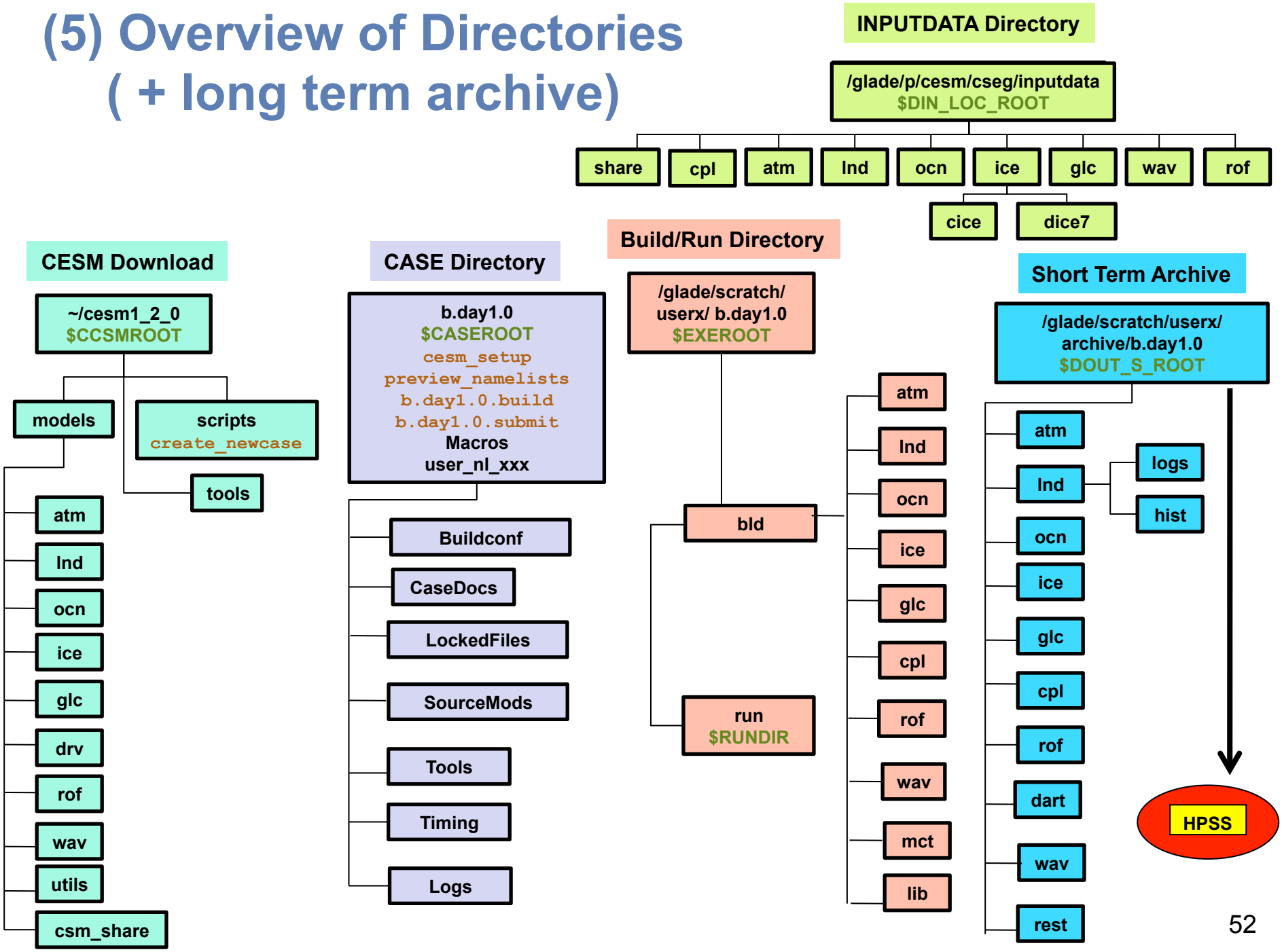
- Why?
  - Migrates output data away from a possibly volatile $DOUT_S_ROOT into a permanent long-term storage area
  - Long term archiving script moves data conveniently and in parallel

- To turn on short term archiving (default is on)
  - Set DOUT_S to TRUE in env_run.xml

- To turn on long term archiving (default is off)
  - Set DOUT_L_MS to TRUE and set DOUT_L_MSROOT in env_run.xml
  - Causes run script to automatically submit a long term archiver job (*.l_archive) at the end of every successful run.

- Long term archiver
  - Moves data from the short term archive directory to a long term archiving system (e.g. HPSS) - if one exists
  - Runs in batch on one processor
  - Can run in parallel with a production job; will not interfere

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



More Information/Getting Help

- **Model User Guides (please provide feedback)**
  - http://www.cesm.ucar.edu/models/cesm1.2/
  - **CESM Users Guide and Web-Browseable code reference**
  - **CAM, CLM, POP2, CICE, Data Model, RTM, and CPL7 Users Guides**
- **CESM Bulletin Board/Discussion Forums**
  - http://bb.cgd.ucar.edu/
  - **Facilitate communication among the community**
  - **Ask questions, look for answers – *all user questions and problems should be posted here***
  - **Many different topics**
- **CESM Release Page Notes**
  - http://www.cesm.ucar.edu/models/cesm1.2/tags/
  - **Notes significant bugs or issues as they are identified**
- **Model output is available on the Earth System Grid**
  - http://www.earthsystemgrid.org

Thank You!

Appendix

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

The NESL Mission is:
To advance understanding of weather, climate, atmospheric composition and processes;
To provide facility support to the wider community; and,
To apply the results to benefit society.

NCAR is sponsored by the National Science Foundation

Appendix A: Steps, Review and How to Undo previous steps

Steps	How to Undo or Change	Associated xml Files
create_newcase	rm –rf $CASEROOT and rerun	env_case.xml
cesm_setup	cesm_setup –clean	env_mach_pes.xml
$CASE*.build	$CASE*.clean_build	env_build.xml, Macros.*
$CASE*.submit	rerun $CASE*.submit	env_run.xml
short term archive	set DOUT_S to False	env_run.xml
$CASE*.l_archive	set DOUT_L_MS to False	env_run.xml

Appendix B: Production Runs

- **Verify**
  - **Setup 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 to $STOP_OPTION and REST_N to $STOP_N to get a restart at end of run.**
- **When initial run is complete**
  - **Set CONTINUE_RUN to TRUE in env_run.xml. This puts the model in restart mode, and the model will start again from the last restart set.**
  - **Reset STOP_N to a larger value if appropriate.**
  - **Leave REST_OPTION==$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.**
- **"cesm_setup –clean" removes the batch run script for target machines. Macros and user_nl_xxx files are never removed by this command. You must remove them manually. A history of your build and machine settings are saved to the PESetupHist subdirectory which is created in your 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, 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.**
- **If a run does NOT complete properly, the stdout file often produces an error message like "Model did not complete – see …/cesm.log…". That cesm log file will often contain a relevant error message, especially in CESM 1.2. Otherwise, all the log files, and stdout/stderr for the job, 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.**

Appendix D: Porting – Machines Directory

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

```
CESM1_2/scripts>ls -l
total 2944
drwxr-sr-x 5 jshollen cseg 131072 May  7 13:53 .
drwxr-sr-x 6 jshollen cseg 131072 May  7 13:53 ..
drwxr-sr-x 8 jshollen cseg 131072 May  7 13:53 ccsm_utils
-rw-r--r-- 1 jshollen cseg 581940 May  7 13:53 ChangeLog
-rwxr-xr-x 1 jshollen cseg  19229 May  7 13:53 create_clone
-rwxr-xr-x 1 jshollen cseg  81134 May  7 13:53 create_newcase
-rwxr-xr-x 1 jshollen cseg  54590 May  7 13:53 create_test
drwxr-sr-x 5 jshollen cseg 131072 May  7 13:53 doc
-rwxr-xr-x 1 jshollen cseg   1255 May  7 13:53 link_dirtree
-rwxr-xr-x 1 jshollen cseg  12701 May  7 13:53 query_tests
-rw-r--r-- 1 jshollen cseg   2345 May  7 13:53 README
-rw-r--r-- 1 jshollen cseg   1113 May  7 13:53 sample_pes_file.xml
drwxr-sr-x 6 jshollen cseg 131072 May  7 13:53 .svn
-rw-r--r-- 1 jshollen cseg    203 May  7 13:53 SVN_EXTERNAL_DIRECTORIES
```

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

•Detailed instructions necessary to port CESM to different machines can be found in the User's Guide.

•Porting steps have changed since the last release.

•We highly recommend you refer to the User's Guide. For further help, contact us through email or one of the discussion forums.

http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/c1719.html

Appendix E: Timing

- **env_mach_pes.xml sets the component pe layout, to change it**
  - **Modify env_mach_pes.xml**
  - **Clean case and setup again**

    b.day1.0> ./cesm_setup –clean

    b.day1.0> ./cesm_setup
  - **Clean and rebuild executables**

    b.day1.0> ./b.day1.0.clean_build

    b.day1.0> ./b.day1.0. build
  - **Resubmit**

    b.day1.0> b.day1.0.submit
- **Timing Files**
  - **See b.day1.0/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 b.day1.0/timing/ccsm_timing.b.day1.0.* file for throughput and load balance (next slide)**
  - **See b.day1.0/timing/ccsm_timing_stats.* for individual rawer model timing output**
  - **Check log file: *b.day1.0>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'**

b.day1.0>cat timing/ccsm_timing.b.day1.0.100519-210440

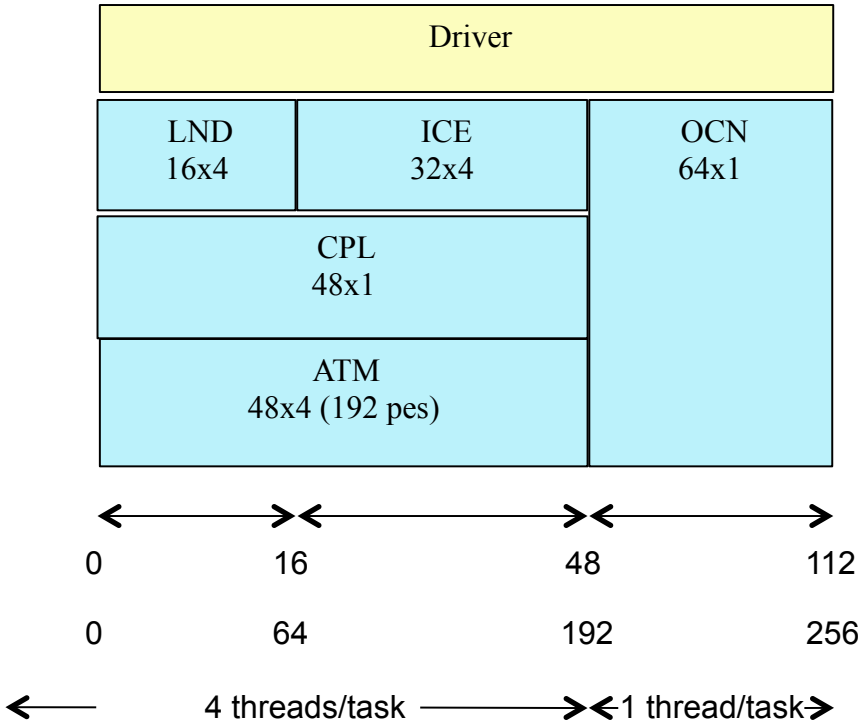
```
 component       comp_pes    root_pe   tasks  x threads instances (stride)
  ---------        ------     -------   ------   ------  ---------  ------
  cpl = cpl        120          0        60     x 2       1       (1     )
  glc = sglc       120          0        60     x 2       1       (1     )
  wav = swav       120          0        60     x 2       1       (1     )
  lnd = clm        60           0        30     x 2       1       (1     )
  rof = rtm        60           0        30     x 2       1       (1     )
  ice = cice       60           0        30     x 2       1       (1     )
  atm = cam        120          0        60     x 2       1       (1     )
  ocn = pop2       60           60       30     x 2       1       (1     )

  total pes active           : 180
  pes per node               : 16
  pe count for cost estimate : 96

  Overall Metrics:
    Model Cost:             149.33   pe-hrs/simulated_year
    Model Throughput:        15.43   simulated_years/day

    Init Time   :      31.609 seconds
    Run Time    :      76.709 seconds       15.342 seconds/day
    Final Time  :       0.032 seconds

<snippet>


    TOT Run Time:      76.709 seconds       15.342 seconds/mday
    LND Run Time:       4.261 seconds        0.852 seconds/mday
    ROF Run Time:       0.632 seconds        0.126 seconds/mday
    ICE Run Time:      10.565 seconds        2.113 seconds/mday
    ATM Run Time:      56.801 seconds       11.360 seconds/mday
    OCN Run Time:       8.299 seconds        1.660 seconds/mday
    GLC Run Time:       0.000 seconds        0.000 seconds/mday
    WAV Run Time:       0.000 seconds        0.000 seconds/mday
    CPL Run Time:       3.807 seconds        0.761 seconds/mday
    CPL COMM Time:     37.395 seconds        7.479 seconds/mday     5
```

Tasks and Threads

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**

A SIMPLE 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"  />
```

112 MPI Tasks

256 Hardware Processors

Appendix F: Testing

- **create_production_test**
  - **Automatically creates a production restart test for the current case**
  - **The test case is created in a parallel directory and called <current case>_<testname>**

- **create_production_test - help explains usage and produces a list of available test types, i.e. <testname>**

- **To use:**

  cases/b.day1.0> ./create_production_test -testname ERT
  cases/b.day1.0> cd b.day1.0.ERT
  cases/b.day1.0_ERT > ./b.day1.0_ERT.build
  cases/b.day1.0_ERT > ./b.day1.0_ERT.submit
  cases/b.day1.0_ERT> cat TestStatus

```
cases/b.day1.0_ERT>  cat TestStatus
                >  PASS b.day1.0_ERT
                    >  PASS b.day1.0_ERT.memleak
```