

Porting CESM 1.2.2

Jim Edwards

CESM Software Engineering Group

Note:

Porting CESM can be a difficult task which may require knowledge of the UNIX operating system, building code with gmake and cmake, scripting with csh, perl and python and submitting and running MPI jobs.

Machine Prerequisites

→ Required:

- ◆ UNIX style operating system
- ◆ csh, sh and perl (with XML:libXML)
- ◆ subversion client
- ◆ Fortran and C compilers (F2003 compliant)
- ◆ MPI
- ◆ NetCDF
- ◆ GNU make
- ◆ Cmake

Machine Prerequisites (cont)

→ Optional

- ◆ PNetCDF
- ◆ ESMF
- ◆ Trillinos
- ◆ LAPACK

Two approaches to porting:

1. Find a CESM supported machine similar to your own and copy the configuration to your new machine definition.
(works very well for Cray systems)

(see <http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/modelnl/machines.html>)

2. Create a case using userdefined as the machine name. Get it working then create the new machine definition.

Creating a userdefined case

```
./create_newcase -case port1 -res f19_g16 -mach  
userdefined -compset X
```

```
cd port1
```

```
./cesm_setup
```

Important Files for Porting:

- **scripts/ccsm_utils/Machines**
 - config_machines.xml => env_build.xml, env_run.xml
 - config_compilers.xml => Macros
 - config_pes.xml => env_mach_pes.xml
 - env_mach_specific.{machine} => env_mach_specific
 - mkbatch.{machine} => Tools/mkbatch.{machine}
- *All files in red are in \$CASEROOT*

config_machines.xml

```
<machine MACH="userdefined">
  <DESC>User Defined Machine</DESC>          <!-- can be anything -->
  <OS>USERDEFINED_required_macros</OS>      <!-- LINUX,Darwin,CNL,AIX,BGL,BGP,BGQ -->
  <COMPILERS>intel,ibm,pgi,pathscale,gnu,cray,lahey</COMPILERS> <!-- intel,ibm,pgi,gnu,cray,nag -->
  <MPILIBS>openmpi,mpich,mpt,mpt,ibm,mpi-serial</MPILIBS> <!-- openmpi, mpich, ibm, mpi-serial -->
  <RUNDIR>USERDEFINED_required_build</RUNDIR> <!-- complete path to the run directory -->
  <EXEROOT>USERDEFINED_required_build</EXEROOT> <!-- complete path to the build directory -->
  <DIN_LOC_ROOT>USERDEFINED_required_build</DIN_LOC_ROOT> <!-- complete path to the inputdata dir -->
  <DIN_LOC_ROOT_CLMFORC>USERDEFINED_optional_build</DIN_LOC_ROOT_CLMFORC>
  <DOUT_S>FALSE</DOUT_S>                     <!-- logical for short term archiving -->
  <DOUT_S_ROOT>USERDEFINED_optional_run</DOUT_S_ROOT> <!-- complete path to a short term archiving dir -->
  <DOUT_L_MSROOT>USERDEFINED_optional_run</DOUT_L_MSROOT> <!-- complete path to a long term archiving directory -->
  <CCSM_BASELINE>USERDEFINED_optional_run</CCSM_BASELINE><!--where the cesm testing scripts write and read baseline results -->
  <CCSM_CPRNC>USERDEFINED_optional_test</CCSM_CPRNC> <!-- path to the cprnc tool used to compare netcdf history files in testing -->
  <BATCHQUERY>USERDEFINED_optional_run</BATCHQUERY>
  <BATCSUBMIT>USERDEFINED_optional_run</BATCSUBMIT>
  <SUPPORTED_BY>USERDEFINED_optional</SUPPORTED_BY>
  <GMAKE_J>1</GMAKE_J>
  <MAX_TASKS_PER_NODE>USERDEFINED_required_build</MAX_TASKS_PER_NODE>
</machine>
```


Run cesm_setup on userdefined

ERROR: must set xml variable OS to generate Macros file

ERROR: must set xml variable NTASKS_CPL to build the model

ERROR: must set xml variable NTASKS_ROF to build the model

ERROR: must set xml variable NTASKS_ICE to build the model

ERROR: must set xml variable CESMSCRATCHROOT to build the model

ERROR: must set xml variable NTASKS_WAV to build the model

ERROR: must set xml variable MPILIB to build the model

ERROR: must set xml variable NTASKS_OCN to build the model

ERROR: must set xml variable NTASKS_GLC to build the model

ERROR: must set xml variable RUNDIR to build the model

ERROR: must set xml variable DIN_LOC_ROOT to build the model

ERROR: must set xml variable COMPILER to build the model

ERROR: must set xml variable EXEROOT to build the model

ERROR: must set xml variable MAX_TASKS_PER_NODE to build the model

ERROR: must set xml variable NTASKS_LND to build the model

ERROR: must set xml variable NTASKS_ATM to build the model

Correct above and issue cesm_setup again

How do you resolve this?

Use xmlchange to set variables correctly

```
./xmlchange OS=Linux
```

```
./xmlchange Compiler=nag
```

```
./xmlchange EXEROOT="\$TMPDIR/\$USER/\$CASE/bld"
```

What is the Macros file?

- Sets compiler and machine specific flags needed for building
- Generated from `config_compilers.xml`
- Acts as an include file for gnu Makefile

```
#
# Makefile Macros generated from cesm1_2_2/scripts/ccsm_utils/Machines/config_compilers.xml using
# COMPILER=intel
# OS=Linux
# MACH=userdefined
CPPDEFS+= -DFORTRANUNDERSCORE -DNO_R16 -DLinux -DCPRINTTEL
SLIBS+=$(shell $(NETCDF_PATH)/bin/nc-config --flibs)
CFLAGS:= -O2 -fp-model precise
FFLAGS:= -fp-model source -convert big_endian -assume byterecl -ftz -traceback -assume realloc_lhs
ifeq ($(DEBUG), TRUE)
    FFLAGS += -O0 -g -check uninit -check bounds -check pointers -fpeo
endif
ifeq ($(DEBUG), FALSE)
    FFLAGS += -O2
endif
ifeq ($(compile_threaded), true)
    LDFLAGS += -openmp
    CFLAGS += -openmp
    FFLAGS += -openmp
endif
```

What does config_compilers.xml look like?

```
<compiler COMPILER="intel">
  <!-- http://software.intel.com/en-us/articles/intel-composer-xe/ -->
  <ADD_CPPDEFS> -DFORTRANUNDERSCORE -DNO_R16</ADD_CPPDEFS>
  <ADD_CFLAGS compile_threaded="true"> -openmp </ADD_CFLAGS>
  <ADD_FFLAGS compile_threaded="true"> -openmp </ADD_FFLAGS>
  <ADD_LDFLAGS compile_threaded="true"> -openmp </ADD_LDFLAGS>
  <FREEFLAGS> -free </FREEFLAGS>
  <FIXEDFLAGS> -fixed -132 </FIXEDFLAGS>
  <ADD_FFLAGS DEBUG="TRUE"> -O0 -g -check uninit -check bounds -check pointers -fpe0 </ADD_FFLAGS>
  <ADD_FFLAGS DEBUG="FALSE"> -O2 </ADD_FFLAGS>
  <FFLAGS> -fp-model source -convert big_endian -assume byterecl -ftz -traceback -assume realloc_lhs </FFLAGS>
  <CFLAGS> -O2 -fp-model precise </CFLAGS>
  <FFLAGS_NOOPT> -O0 </FFLAGS_NOOPT>
  <FC_AUTO_R8> -r8 </FC_AUTO_R8>
  <SFC> ifort </SFC>
  <SCC> icc </SCC>
  <SCXX> icpc </SCXX>
  <MPIFC> mpif90 </MPIFC>
  <MPICC> mpicc </MPICC>
  <MPICXX> mpicxx </MPICXX>
  <CXX_LINKER>FORTRAN</CXX_LINKER>
  <CXX_LDFLAGS> -cxxlib </CXX_LDFLAGS>
  <SUPPORTS_CXX>TRUE</SUPPORTS_CXX>
  <HAS_F2008_CONTIGUOUS>TRUE</HAS_F2008_CONTIGUOUS>
</compiler>
```

more on config_compilers.xml

```
<compiler MACH="goldbach">  
  <MPI_LIB_NAME MPILIB="openmpi"> mpi</MPI_LIB_NAME>  
  <LAPACK_LIBDIR> /usr/lib64 </LAPACK_LIBDIR>  
  <ADD_SLIBS>$(shell $(NETCDF_PATH)/bin/nf-config --flibs)</ADD_SLIBS>  
</compiler>  
  
<compiler COMPILER="intel" MACH="goldbach">  
  <ADD_LDFLAGS> -Wl,-rpath,$(NETCDF_PATH)/lib </ADD_LDFLAGS>  
  <ADD_LDFLAGS> -Wl,-rpath,$(COMPILER_PATH)/lib/intel64 </ADD_LDFLAGS>  
</compiler>
```

mkbatch.{machine}

- Three sections (Phases)
 - **set_batch** - creates the (PBS,LSF,COBAL) stanza of the run script
 - **set_exe** - creates the job launch stanza of the run script
 - **set_larch** - creates the Long Term archiving stanza of the run script
- Written in csh (migrating to perl in upcoming releases)

Frequently encountered setup issues:

- Can't find or download inputdata
 - Invalid DIN_LOC_ROOT or no write permission
 - No svn access to data server

DO NOT DOWNLOAD all the inputdata! When properly configured the model will retrieve just what you need.

Frequently encountered build issues:

- Can't find compiler(s) or MPI libraries
 - Build fails in MCT
- Can't find NetCDF or Cmake
 - Build fails in PIO

Resources:

The Users Manual: <http://www.cesm.ucar.edu/models/cesm1.2/cesm/doc/usersguide/>

The bulletin board: <http://forum.cgd.ucar.edu>