NCAR UCAR

# Porting CESM 2.1.5

Jim Edwards
CESM Software Engineering Group

NCAR is sponsored by National Science Foundation

CGD Climate & Global Dynamics a Laboratory of NCAR

7/11/2025

# When and Why You Might Need to Port CESM

- **CESM must be built and tested on each system where it is used**
  - Different clusters, OS versions, and compilers require different configurations
- **You might need to port CESM if:**
  - You're using a new HPC system or workstation
  - The system isn't already defined in `config_machines.xml`
  - You're using a new compiler or MPI library
- **Goals of porting:**
  - Get CESM to build and run
  - Ensure correctness and performance

NCAR is sponsored by National Science Foundation

# Options for porting CESM 2.1.5

- **Use predefined machine definitions**
  (e.g., `homebrew`, `centos7-linux`)
  - Good for Mac/CentOS with GNU compilers
  - May require root access to install dependencies
- **Local definition in** `$HOME/.cime`
  - Ideal for personal use
  - Fully customizable and non-intrusive
- **Submit a machine definition to CIME**
  - For shared HPC systems
  - Requires GitHub familiarity and thorough testing

# CESM Prerequisites

- Linux or MAC/OS operating system
- cmake
- gnu make
- HDF5 (1.12.3 recommended)
- NetCDF (4.9.3)
- PNetCDF (1.14.0) - optional
- C/Fortran/C++ Compilers: Intel recommended, gnu, nvhpc, nag
- lapack & blas math libraries (often supplied with compiler)
- ESMF (required for cesm3, WACCM)

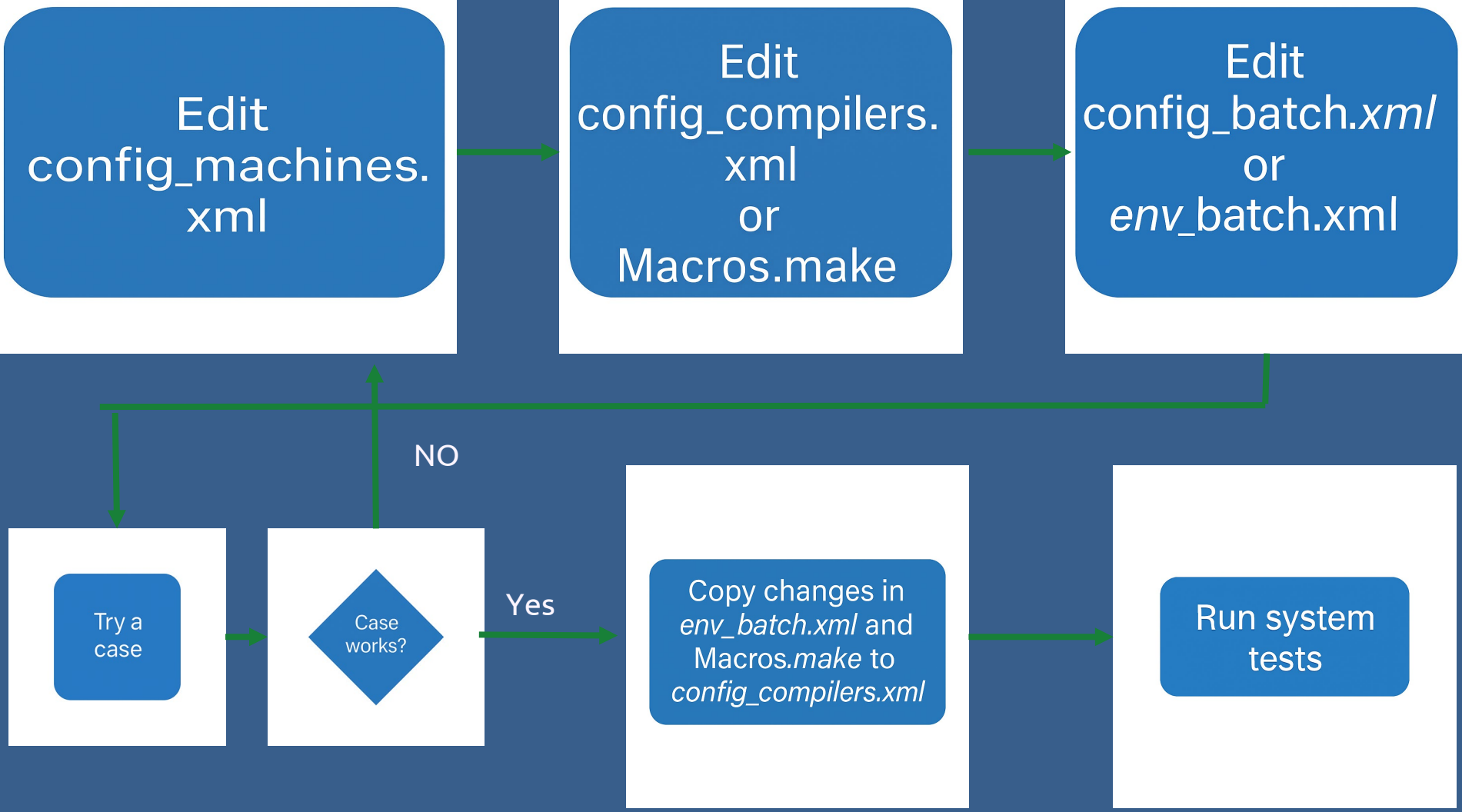
NCAR is sponsored by National Science Foundation

# The $HOME/.cime directory

When you use a CIME model it will look for a directory $HOME/.cime

You may put several files in that directory for CIME to use.

1. config: custom settings used by the python scripts
2. config_machines.xml: custom machine description
3. config_compilers.xml: custom compiler instructions
4. config_batch.xml: custom batch system instructions


Edit config_machines.xml
Edit config_compilers.xml or Macros.make
Edit config_batch.xml or env_batch.xml
NO
Try a case
Case works?
Yes
Copy changes in env_batch.xml and Macros.make to config_compilers.xml
Run system tests

# Things that you set in config_machines.xml

- Path to default case, build and run directories
- Path to inputdata
- Supported compilers and mpi libraries
- System description
  - Operating system
  - Number of tasks per shared memory node
  - nodename regex for system identification
- Type of batch system (pbs, slurm, lsf, etc)
- Support for (environment or Lmod) modules if used
- Environment variable settings

# Things that you set in config_compilers.xml

- Compiler flags that should be applied to all files in the build
- Linker flags and external library links

Note that this file is hierarchical. There is a general section for each compiler. This section can be overridden by subsequent sections with any combination of the following attributes:

- COMPILER
- OS
- MACH
- DEBUG
- MODEL
- MPILIB

# Things that you set in config_batch.xml

- queue names and parameters (wall clock time, minimum and maximum job size)
- details of deviations from standard queue system definitions
- directives (batch specific commands added to submitted scripts)
- arguments (added to the job submit command line)

Again this file is hierarchical, general settings can be overridden in machine specific sections.

# config_workflow.xml

This file allows for a simple single run workflow to be defined. The default workflow consists of running the case and subsequently the case archive step.

Workflows can be extended to add things like:

- postprocessing (diagnostics, time-series generation, file compression, etc)
- iterative run feedback and adjustments

# scripts_regression_tests.py

- A porting test for basic model functionality.
- requires that you build and install cprnc
- optionally uses pylint (install using pip or conda)
- Run from cime/scripts/tests

NCAR is sponsored by
National Science Foundation

# The CESM UltraFast Ensemble Consistency Test

With this test you will conduct three model runs of 9 timesteps each.

Postprocess the output files to add metadata about the origin of the run.

Upload to http://www.cesm.ucar.edu/models/cesm2/verification/

Your runs are then compared to an ensemble of about 300 members.

Pass or Fail is determined by whether your runs fit within the bounds of the ensemble.

See http://www.cesm.ucar.edu/models/cesm2/python-tools/ for details

# Resources for help

CGD Forum: http://forum.cgd.ucar.edu/

ESCOMP development (CESM): https://github.com/ESCOMP/CESM

ESMCI development (CIME): https://github.com/ESMCI/cime

# Changes in CESM3

- MCT coupler/driver has been replaced with ESMF based CMEPS driver and mediator
- New CDEPS data model components
- config_compilers.xml replaced with cmake modules
- More interchangeable component options
- Much easier to introduce new grids
  - offline land fraction and mapping files are no longer required - this reduces the number of required grid files from ~25 to 4
- machines directory has been split into subdirectories (one for each machine)
- cime has been split into several repositories, what remains in cime is primarily the python workflow code.
  - porting information is now in ccs_config
  - cmeps and cdeps are each their own repos
  - share code is now its own repository

# CESM on your Laptop

With containers we can provide the community with *ready-to-run* CESM software - porting is much simpler!

They're also *portable* across Mac, Windows and Linux.

One popular variant ("CESM-Lab") includes a Jupyter Lab environment and a basic tutorial.

Great for laptops / desktops, and thus simple or low-res models. HPC version for clusters is being developed.

# CESM on the Cloud

If you don't have a local cluster, CESM can also run on the cloud - on AWS now, and Azure soon.

Also fully preconfigured; no porting needed.

However, cloud *costs* are considerably higher than academic HPC systems, which limits use cases.

Public gateway / tool coming later this year.

# Questions / Comments?

jedwards@ucar.edu

NCAR is sponsored by
National Science Foundation