NorESM user workshop 2021

15 - 17 Nov. 2021

Alok K. Gupta, Yanchun He, Mats Bentsen, Ada Gjermundsen, Tomas Torsvik

Forskningsrådet

This work was supported by the Norwegian Research Council INES (grant no. 270061)

Agenda

For this workshop we will demonstrate how to set up, build and run the NorESM model. We will go through the process two times, first time with a minimal setup and second time focusing on different options for each step.

1. Basic steps to set up a NorESM experiment, build the model and start a simulation run (Ada)
2. NorESM model structure (Tomas)
3. HPC resources provided by Sigma2 (Tomas)
4. Creating a new case (Tomas)
5. Options for building the case (Ada)
6. Options for running the case (Ada)
7. NorESM log files (Tomas)
8. Use a github fork for NorESM model development (Tomas)
9. Including your own code contributions for a case (Tomas)

Web services:

- Research Data Archive : https://archive.norstore.no/
- Diagnostic output : http://ns2345k.web.sigma2.no/
- ESGF node : https://noresg.nird.sigma2.no/thredds/
- NorESM git repository : https://github.com/NorESMhub
- NIRD tootkit : https://apps.sigma2.no/nird
- NorESM documentation : https://noresm-docs.readthedocs.io/en/latest/
- NorESM inputdata Server : https://www.noresm.org/inputdata

Resources for NorESM workshop

Reserved queue on Betzy for NorESM user workshop

During the user workshop we have access to a reserved queue on Betzy. The queue reservation is in place until 16 Nov 2021 at 23:59.

ReservationName = noresm
Accounts = nn9560k
PartitionName = normal
Nodes=b[4149-4162,4169-4171] NodeCnt=17 CoreCnt=2176
StartTime=2021-11-15T08:00:00 EndTime=2021-11-17T00:00:00 Duration=1-16:00:00

You can use this reservation by specifying it in sbatch as

```
#SBATCH --res=noresm
```

or in interactive command line as

```
--reservation=noresm
```

NIRD storage location

Output from NorESM runs can be copied to NS2345K project location on Nird:

1. Log in to Nird: `ssh <username>@login.nird.sigma2.no`

2. `cd /projects/NS2345K/workshop2021`

3. Create a subfolder for your own output files, e.g.
   `mkdir <username>`

4. On Betzy : Copy output from Betzy to Nird
   `scp -r <path/to/noresm/output> <username>@login.nird.sigma2.no:/projects/NS2345K/workshop2021/<username>/`

NOTE: The workshop2021 directory will be deleted from NS2345K shortly after the end of the workshop.

Downloading the NorESM code (Ada)

- An recipe for downloading the CMIP6 version of NorESM

Downloading the NorESM code

Please deactivate your conda environment if you have one on betzy:
$ conda deactivate

Recipe:

1. Log on to betzy
2. Make a directory which will contain the NorESM code and the experiments
3. Enter the folder
4. Make a directory called cases. Here you will store all your NorESM experiments.
5. Get the NorESM code from GitHub
6. Enter the noresm repository
7. Checkout the CMIP6 version of NorESM
8. Get the corresponding code for the model component and cime

1. **$ ssh username@betzy.sigma2.no**
2. **$ mkdir NorESM**
3. **$ cd NorESM**
4. **$ mkdir cases**
5. **$ git clone https://github.com/NorESMhub/NorESM.git noresm2.0**
6. **$ cd noresm2.0**
7. **$ git checkout release-noresm2.0.5**
8. **$ ./manage_externals/checkout_externals**

Documentation: https://noresm-docs.readthedocs.io/en/latest/access/download_code.html

All the steps needed for downloading the CMIP6 version of NorESM2

```
[adagj@login-1.BETZY ~]$ mkdir NorESM
[adagj@login-1.BETZY ~]$ cd NorESM
[adagj@login-1.BETZY ~/NorESM]$ mkdir cases
[adagj@login-1.BETZY ~/NorESM]$ git clone https://github.com/NorESMhub/NorESM.git noresm2.0
Cloning into 'noresm2.0'...
remote: Enumerating objects: 8270, done.
remote: Counting objects: 100% (2472/2472), done.
remote: Compressing objects: 100% (1229/1229), done.
remote: Total 8270 (delta 1574), reused 1956 (delta 1231), pack-reused 5798
Receiving objects: 100% (8270/8270), 41.16 MiB | 5.12 MiB/s, done.
Resolving deltas: 100% (5230/5230), done.
[adagj@login-1.BETZY ~/NorESM]$ cd noresm2.0/
[adagj@login-1.BETZY ~/NorESM/noresm2.0]$ git checkout release-noresm2.0.5
Note: checking out 'release-noresm2.0.5'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at 133cc12... Merge pull request #279 from DirkOlivie/noresm2
[adagj@login-1.BETZY ~/NorESM/noresm2.0]$ ./manage_externals/checkout_externals
Processing externals description file : Externals.cfg
Checking status of externals: clm, mosart, ww3, cime, cice, pop, cism, rtm, cam, blom,
Checking out externals: clm, mosart, cime, cice, cam, blom,
Processing externals description file : Externals_CLM.cfg
Checking out externals: fates, ptclm,

[adagj@login-1.BETZY ~/NorESM/noresm2.0]$
```

Basic steps to run NorESM (Ada)

- The Newbies Guide

Basic steps to run NorESM: Create case

The **create_newcase** script is an executable python script located in cime/scripts

./create_newcase --case $PATH_TO_cases/$CASENAME --mach $MACHINE --res $RESOLUTION --compset $COMPSET --project $PROJECT --user-mods-dir $USER_MOD_DIRS

Tomas will give a detailed description in the next session

--case $CASENAME -> name of the NorESM experiment you are creating
--mach $MACHINE -> name of the HPC you are using e.g. betzy, fram, nebula
--res $RESOLUTION -> running with 1 (f09_tn14) or 2 degree (f19_tn14) atmosphere/land resolution
--compset $COMPSETNAME -> e.g. piControl (N1850frc2), historical, ssp585, abrupt-4xCO2
--project $PROJECT -> which project provides the cpu hours you are using (e.g. nn9560k = INES)
(--user-mods-dir -> predefined namelists and source mods)

$ cd ~/NorESM/noresm2.0/cime/scripts/

$ ./create_newcase --case ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115 --mach betzy --res f19_tn14 --compset N1850frc2 --project nn9560k --pecount=128

During this workshop we will run NorESM on 1 node = 128 processors

./create_newcase --help will provide you all input options including a description

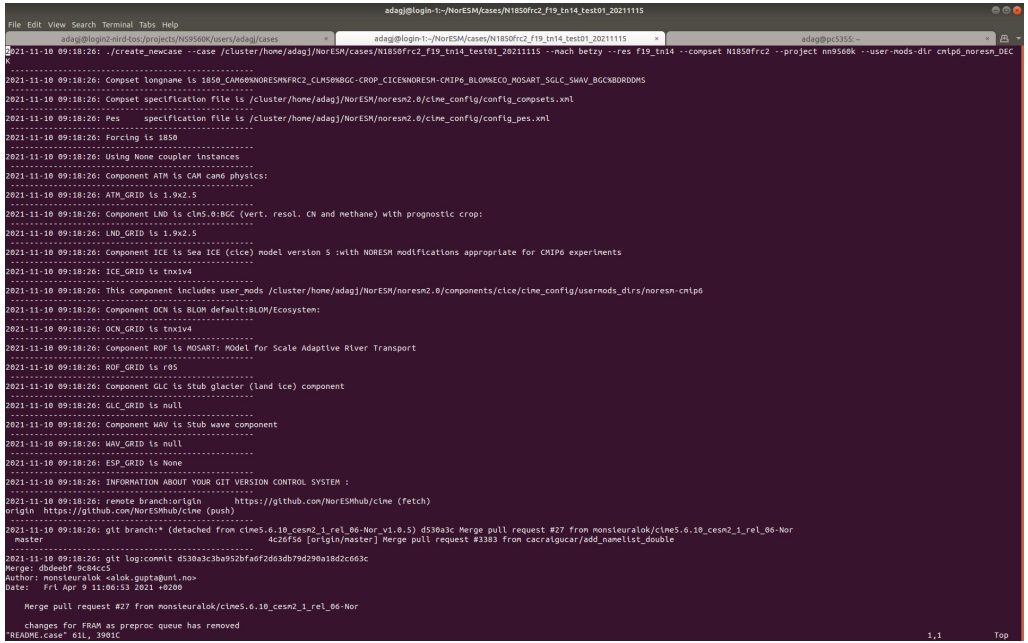
Documentation: https://noresm-docs.readthedocs.io/en/latest/configurations/newbie-guide.html

Basic steps to run NorESM: Create case

```
[adagj@login-1.BETZY ~]$ cd ~/NorESM/noresm2.0/cime/scripts/
[adagj@login-1.BETZY ~/NorESM/noresm2.0/cime/scripts]$ ./create_newcase --case ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115 --mach betzy --res f19_tn14 --compset N1850frc2 --project nn9560k --user-mods-dir
cmip6_noresm_DECK
Compset longname is 1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS
Compset specification file is /cluster/home/adagj/NorESM/noresm2.0/cime/../cime_config/config_compsets.xml
Compset forcing is 1850
ATM component is CAM cam6 physics:
LND component is clm5.0:BGC (vert. resol. CN and methane) with prognostic crop:
ICE component is Sea ICE (cice) model version 5 :with NORESM modifications appropriate for CMIP6 experiments
OCN component is BLOM default:BLOM/Ecosystem:
ROF component is MOSART: MOdel for Scale Adaptive River Transport
GLC component is Stub glacier (land ice) component
WAV component is Stub wave component
ESP component is
Pes     specification file is /cluster/home/adagj/NorESM/noresm2.0/cime/../cime_config/config_pes.xml
Compset specific settings: name is RUN_STARTDATE and value is 0001-01-01
Could not find machine match for 'login-1.betzy.sigma2.no' or 'login-1.betzy.sigma2.no'
Machine is betzy
Pes setting: grid match    is a%1.9x2.5.+l%1.9x2.5.+oi%tnx1v4
Pes setting: machine match is betzy
Pes setting: compset_match is CAM60%NORESM.+CLM50%BGC-CROP.+CICE.+BLOM%ECO
Pes setting: pesize match  is M
Pes setting: grid          is a%1.9x2.5_l%1.9x2.5_oi%tnx1v4_r%r05_g%null_w%null_m%tnx1v4
Pes setting: compset       is 1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS
Pes setting: tasks       is {'NTASKS_ATM': 768, 'NTASKS_ICE': 544, 'NTASKS_CPL': 768, 'NTASKS_LND': 192, 'NTASKS_WAV': 32, 'NTASKS_ROF': 128, 'NTASKS_OCN': 256, 'NTASKS_GLC': 768}
Pes setting: threads     is {'NTHRDS_ICE': 1, 'NTHRDS_ATM': 1, 'NTHRDS_ROF': 1, 'NTHRDS_LND': 1, 'NTHRDS_WAV': 1, 'NTHRDS_OCN': 1, 'NTHRDS_CPL': 1, 'NTHRDS_GLC': 1}
Pes setting: rootpe      is {'ROOTPE_OCN': 768, 'ROOTPE_LND': 0, 'ROOTPE_ATM': 0, 'ROOTPE_ICE': 224, 'ROOTPE_WAV': 192, 'ROOTPE_CPL': 0, 'ROOTPE_ROF': 0, 'ROOTPE_GLC': 0}
Pes setting: pstrid      is {}
Pes other settings: {}
Pes comments: none
 Compset is: 1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS
 Grid is: a%1.9x2.5_l%1.9x2.5_oi%tnx1v4_r%r05_g%null_w%null_m%tnx1v4
 Components in compset are: ['cam', 'clm', 'cice', 'blom', 'mosart', 'sglc', 'swav', 'sesp', 'drv', 'dart']

This is a CESM or NorESM scientifically supported compset at this resolution.

No charge_account info available, using value from PROJECT
No project info available
cesm model version found: release-noresm2.0.5
Batch_system_type is slurm_nor
job is case.run USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
job is case.st_archive USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
 Creating Case directory /cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test01_20211115
This component includes user_mods /cluster/home/adagj/NorESM/noresm2.0/components/cice/cime_config/usermods_dirs/noresm-cmip6
Adding user mods directory /cluster/home/adagj/NorESM/noresm2.0/components/cice/cime_config/usermods_dirs/noresm-cmip6
Adding user mods directory /cluster/home/adagj/NorESM/noresm2.0/cime_config/usermods_dirs/cmip6_noresm_DECK
Adding SourceMod to case /cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/SourceMods/src.cam/preprocessorDefinitions.h
```

README

```
$ cd ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/

$ vi README.case
```



You will find the information about your case in README

Including compset long name, grid files, components, git branch, git commit etc.

README can be very useful if you want to reproduce a case (either your own or somebody else's case)

Basic steps to run NorESM: Set up the case

Enter the case folder:

```
$ cd ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/

$ ./xmlchange NTASKS_OCN=123

$ ./case.setup
```

NOTE!
During this workshop we will run NorESM on 1 node.
Thus, we need to make changes to env_mach_pes.xml before running ./case.setup
You can do so by the use of xmlchange

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/newbie-guide.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiments.html#create-and-configure-a-new-case
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#machine-specific-environment

Basic steps to run NorESM: Set up the case

$ ./case.setup

```
[adagj@login-2.BETZY ~/NorESM/cases]$ cd ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/
[adagj@login-2.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115]$ ./xmlchange NTASKS_OCN=123
[adagj@login-2.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115]$ ./case.setup
Setting resource.RLIMIT_STACK to -1 from (8388608, -1)
/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/env_mach_specific.xml already exists, delete to replace
job is case.run USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
Creating batch scripts
Writing case.run script from input template /cluster/home/adagj/NorESM/noresm2.0/cime/config/cesm/machines/template.case.run
Creating file .case.run
Writing case.st_archive script from input template /cluster/home/adagj/NorESM/noresm2.0/cime/config/cesm/machines/template.st_archive
Creating file case.st_archive
Creating user_nl_xxx files for components and cpl
If an old case build already exists, might want to run 'case.build --clean' before building
You can now run './preview_run' to get more info on how your case will be run
[adagj@login-2.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115]$
```

Basic steps to run NorESM: Build the case

After running ./case.setup you should see your case in the noresm run directory

```
$ ls /cluster/work/users/adagj/noresm/

N1850frc2_f19_tn14_test01_20211115

$ ls /cluster/work/users/adagj/noresm/N1850frc2_f19_tn14_test01_20211115/
  bld  run
```

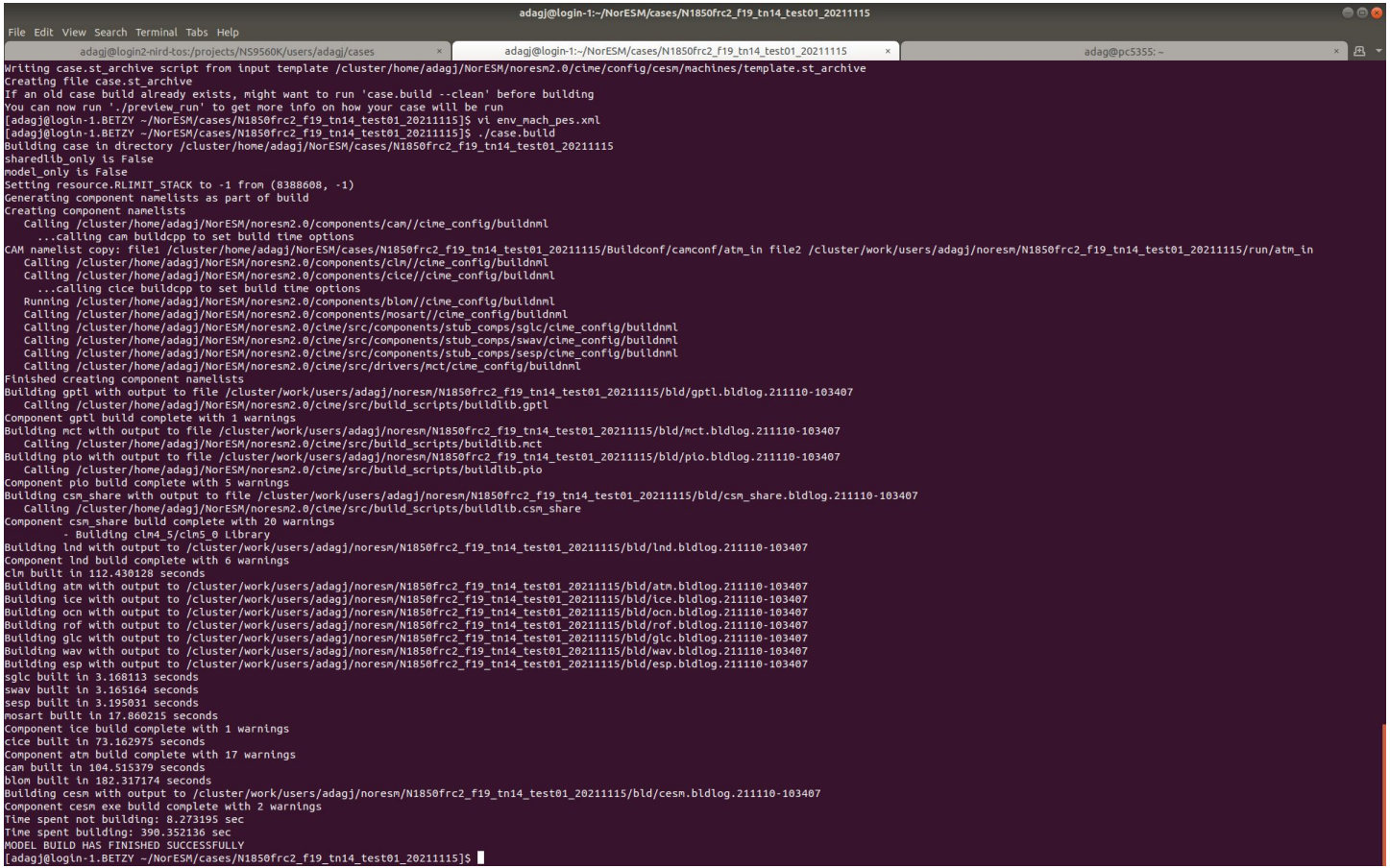
Please use your own username (instead of mine :-)

bld: the build folder
run: the run folder. Here you'll find logs, output data, restart files etc. More on that later ...

```
$ ./case.build
```

Creating an executable **cesm.exe**; found in /cluster/work/users/$USER/noresm/$CASENAME/bld

Basic steps to run NorESM: Building the case



Basic steps to run NorESM: Submitting the case

$ vi env_batch.xml

Please note!
Usually you will not make these changes, because you will run NorESM on several more nodes

In env_batch.xml on line 37 (37G), **change p to q** (i activates insert):

<arg flag="-p" name="$JOB_QUEUE"/> → <arg flag="-q" name="$JOB_QUEUE"/>

And on line 81 (81G), change **queue from normal to devel** (development):

<entry id="JOB_QUEUE" value="normal"> → <entry id="JOB_QUEUE" value="devel">

$ ./case.submit

```
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test03_20211115]$ ./case.submit
File /cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test03_20211115/LockedFiles/env_batch.xml has been modified
  found difference in JOB_QUEUE : case 'devel' locked 'preproc'

env_batch.xml appears to have changed, regenerating batch scripts
manual edits to these file will be lost!

Creating batch scripts
Writing case.run script from input template /cluster/home/adagj/NorESM/noresm2.0/cime/config/cesm/machines/template.case.run
Creating file .case.run
Writing case.st_archive script from input template /cluster/home/adagj/NorESM/noresm2.0/cime/config/cesm/machines/template.st_archive
Creating file case.st_archive
Setting resource.RLIMIT_STACK to -1 from (-1, -1)
Creating component namelists
  Calling /cluster/home/adagj/NorESM/noresm2.0/components/cam//cime_config/buildnml
CAM namelist copy: file1 /cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test03_20211115/Buildconf/camconf/atm_in file2 /cluster/work/users/adagj/noresm/N1850frc2_f19_tn14_test03_20211115/run/atm_in
  Calling /cluster/home/adagj/NorESM/noresm2.0/components/clm//cime_config/buildnml
  Calling /cluster/home/adagj/NorESM/noresm2.0/components/cice//cime_config/buildnml
  Running /cluster/home/adagj/NorESM/noresm2.0/components/blom//cime_config/buildnml
  Calling /cluster/home/adagj/NorESM/noresm2.0/components/mosart//cime_config/buildnml
  Calling /cluster/home/adagj/NorESM/noresm2.0/cime/src/components/stub_comps/sglc/cime_config/buildnml
  Calling /cluster/home/adagj/NorESM/noresm2.0/cime/src/components/stub_comps/swav/cime_config/buildnml
  Calling /cluster/home/adagj/NorESM/noresm2.0/cime/src/components/stub_comps/sesp/cime_config/buildnml
  Calling /cluster/home/adagj/NorESM/noresm2.0/cime/src/drivers/mct/cime_config/buildnml
Finished creating component namelists
Checking that inputdata is available as part of case submission
Setting resource.RLIMIT_STACK to -1 from (-1, -1)
Loading input file list: 'Buildconf/can.input_data_list'
Loading input file list: 'Buildconf/cpl.input_data_list'
Loading input file list: 'Buildconf/blom.input_data_list'
Loading input file list: 'Buildconf/cice.input_data_list'
Loading input file list: 'Buildconf/mosart.input_data_list'
Loading input file list: 'Buildconf/clm.input_data_list'
Check case OK
submit_jobs case.run
Submit job case.run
Submitting job script sbatch --time 00:59:00 -q devel --account nn9560k .case.run --resubmit
Submitted job id is 253504
Submit job case.st_archive
Submitting job script sbatch --time 0:59:00 -q preproc --account nn9560k  --dependency=afterok:253504 case.st_archive --resubmit
Submitted job id is 253505
Submitted job case.run with id 253504
Submitted job case.st_archive with id 253505
```

Monitoring your jobs: some useful commands

squeue: overview of job(s) running and the job id(s)

scontrol: see more details about the job running

scancel: stop job from running

```
$ squeue -u $USER

$ squeue -p $PROJECT

$ scontrol show job $JOBID

$ scancel $JOBID
```

Note! squeue -u $USER: If you don't see your job, it has either finished or crashed!

Monitoring jobs : https://documentation.sigma2.no/jobs/monitoring.html

Hands-on session 1

Download the CMIP6 version of NorESM2:

1. $ ssh username@betzy.sigma2.no
2. $ mkdir NorESM
3. $ cd NorESM
4. $ mkdir cases
5. $ git clone https://github.com/NorESMhub/NorESM.git noresm2.0
6. $ cd noresm2.0
7. $ git checkout release-noresm2.0.5
8. $ ./manage_externals/checkout_externals

Hands-on session 1

Create, setup, build and submit your (first?) NorESM2-LM piControl simulation running on 1 node by repeating these steps:

```
$ cd ~/NorESM/noresm2.0/cime/scripts/

$ ./create_newcase --case ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115 --mach betzy
--res f19_tn14 --compset N1850frc2 --project nn9560k --pecount=128

$ cd ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/

$ ./xmlchange NTASKS_OCN=123

$ ./case.setup

$ ./case.build

$ vi env_batch.xml

$./case.submit
```

In env_batch.xml on line 37 (37G), **change p to q** (i activates insert):

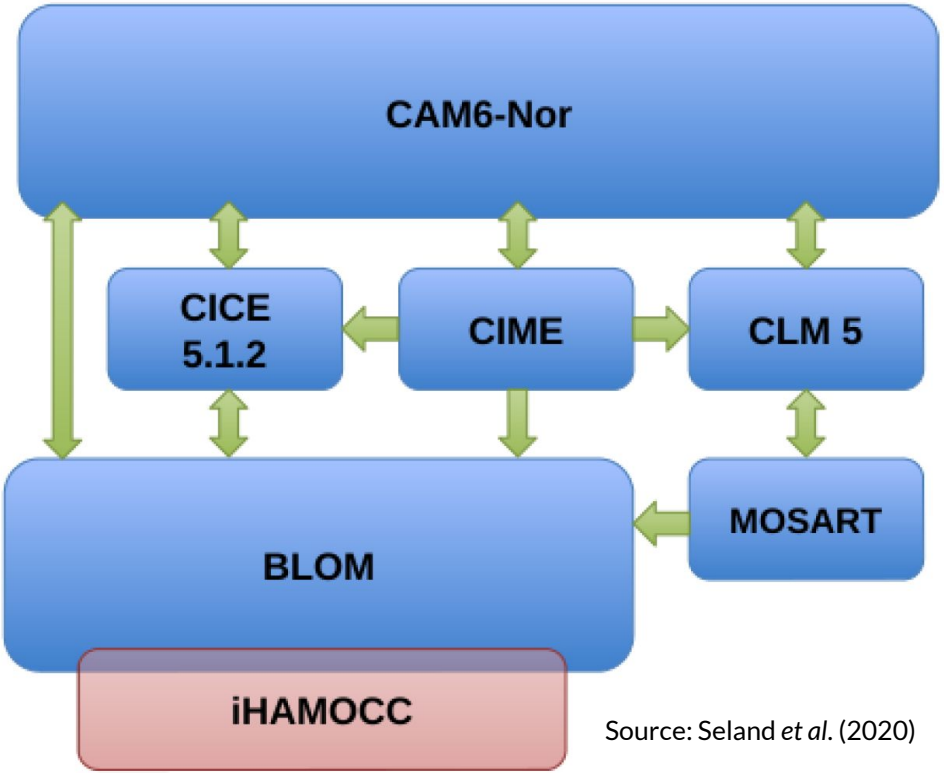
<arg flag="-p" name="$JOB_QUEUE"/> <arg flag="-q" name="$JOB_QUEUE"/>

And on line 81, change queue from normal to devel (development):

<entry id="JOB_QUEUE" value="normal"> <entry id="JOB_QUEUE" value="devel">

NorESM model system (Tomas)

NorESM framework

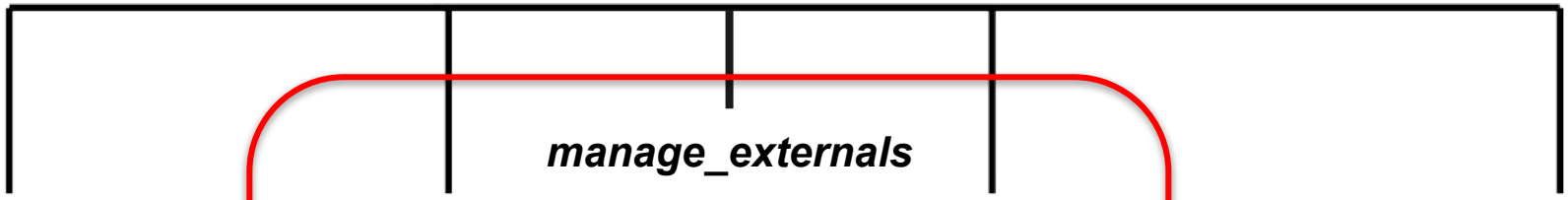


Source: Seland *et al.* (2020)

NorESM2 is based on the second version of the Community Earth System Model, CESM2, and share most of the CESM2 structure, but modifies model component.

- Atmosphere model :
  CAM6-Nor replaces standard CAM
- Atmospheric chemistry:
  OsloAero6
- Ocean model :
  Isopycnic coordinate model BLOM
- Ocean biogeochemical model :
  iHAMOCC
- Sea-ice model:
  Wind drift of snow

NorESM model structure



cime

Most general scripts for creating, submitting, machines related setting, IO code (PIO), data components, coupler code, archiving scripts

doc

NorESM specific information
CMIP6 related key experiments
....

cime_config

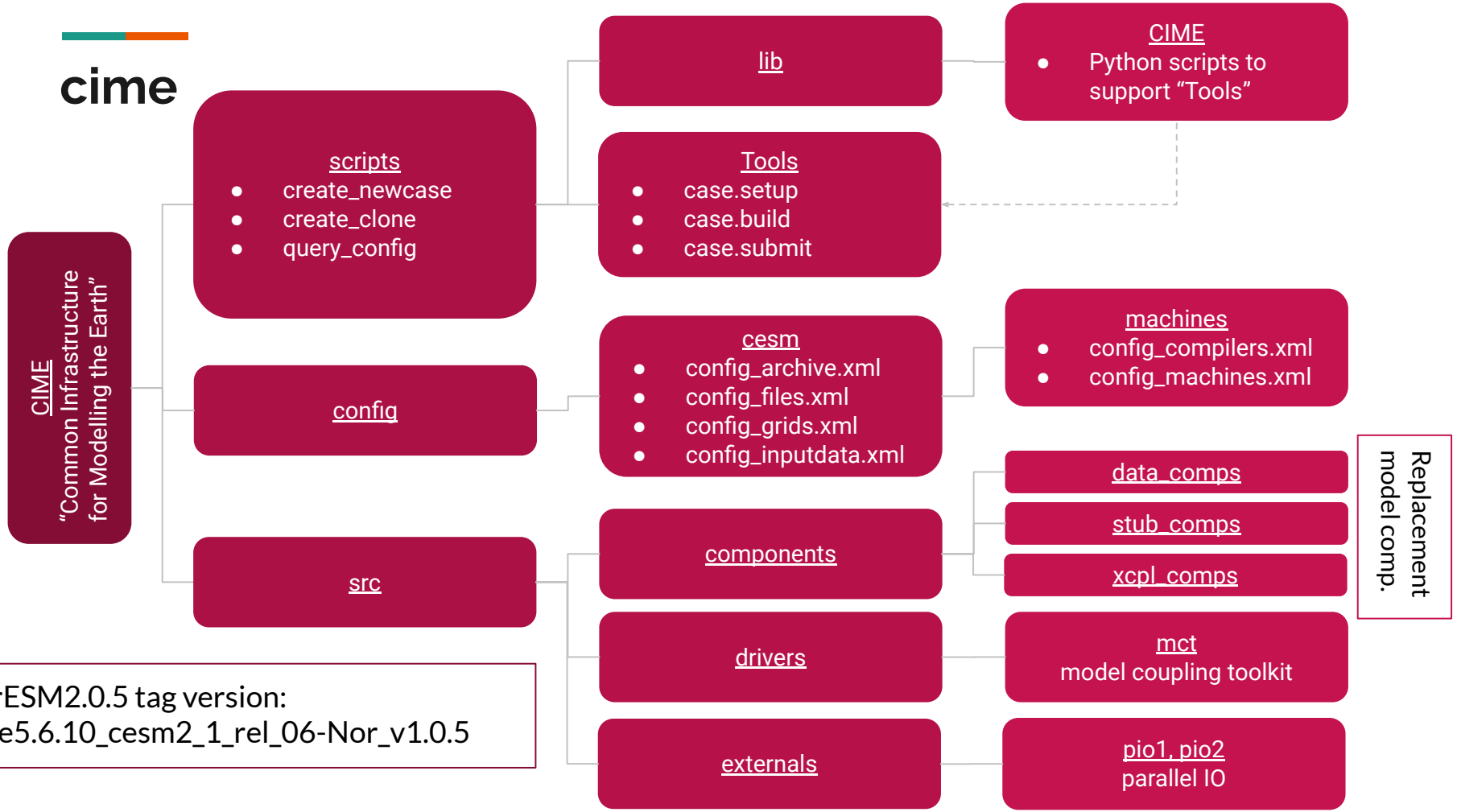
Coupled experiments configurations and processors settings

NorESM repository

components

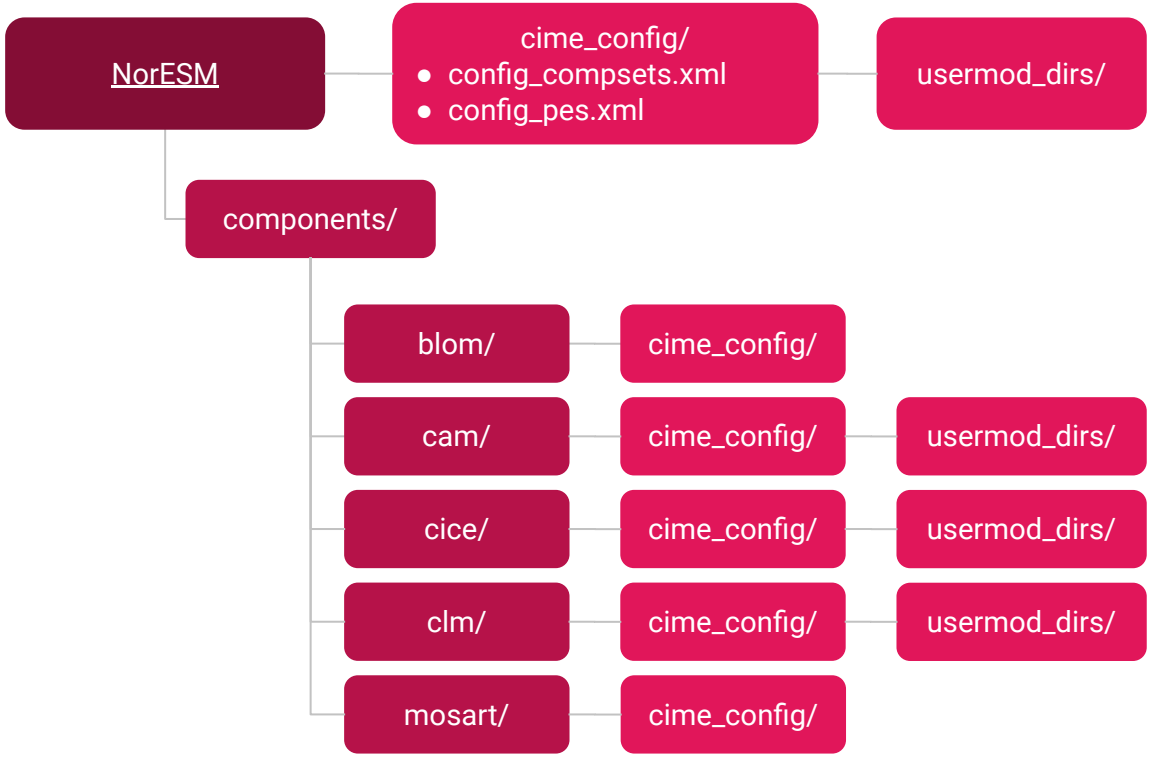
All active components and their default namelist settings , processors and experiments setting when not fully coupled

cime



NorESM2.0.5 tag version:
cime5.6.10_cesm2_1_rel_06-Nor_v1.0.5

cime_config directory



Files in cime_config/ :

config_compsets.xml : compset configuration file

config_pes.xml : processor config for particular compsets

Additional component scripts

buildnml : build component namelist

buildlib : build component library

Fetching the source: *checkout_externals* script

`./manage_externals/checkout_externals`	Fetch model components defined in the `Externals.cfg` file
`./checkout_externals -e [EXTERNALS]`	Fetch model components defined in `[EXTERNALS]` file
`./checkout_externals -S`	Check status of downloaded model components
`./checkout_externals -h`	See all options for `checkout_externals`

Entries from **Externals.cfg** file :

```
[cam]
tag = cam_cesm2_1_rel_05-Nor_v1.0.4
protocol = git
repo_url = https://github.com/NorESMhub/CAM
local_path = components/cam
required = True
```

```
[clm]
tag = release-clm5.0.14-Nor_v1.0.3
protocol = git
repo_url = https://github.com/NorESMhub/ctsm
local_path = components/clm
externals = Externals_CLM.cfg
required = True
```

HPC resources provided by Sigma2 (Tomas)

Sigma2 HPC machines

See Sigma2 documentation for more HPC info: https://documentation.sigma2.no/hpc_machines/hardware_overview.html

Sigma2 provides 3 HPC systems:

- Betzy for large parallel jobs (Normal queue: 4-512 nodes, 4 days max walltime)
- Fram for intermediate parallel jobs (Normal queue: 1-32 nodes, 7 days max walltime)
- Saga for serial or single node jobs (Not configured for NorESM)

HPC job types: https://documentation.sigma2.no/jobs/choosing_job_types.html

	Fram	Betzy
System	Lenovo NeXtScale nx360	BullSequana XH2000
CPU type	Intel E5-2683v4 ; 2.1 GHz	AMD Epyc 7742 ; 2.25GHz
Nodes / Cores	1006 / 32256	1344 / 172032
(core / mem) per node	32 / 64 GB	128 / 256 GB
	largemem: 8x 512GB, 2x 6TB	16 Nvidia A100 GPUs

Fram and Betzy storage areas

Directory	Alias	Purpose
/cluster/home/$USER	$HOME	User data
/custer/work/users/$USER	$USERWORK	Staging and job data
/custer/work/jobs/$SLURM_JOB_ID	$SCRATCH	Per-job data
/cluster/projects/<project_name>		Project data
/cluster/shared/<folder_name>		Shared data

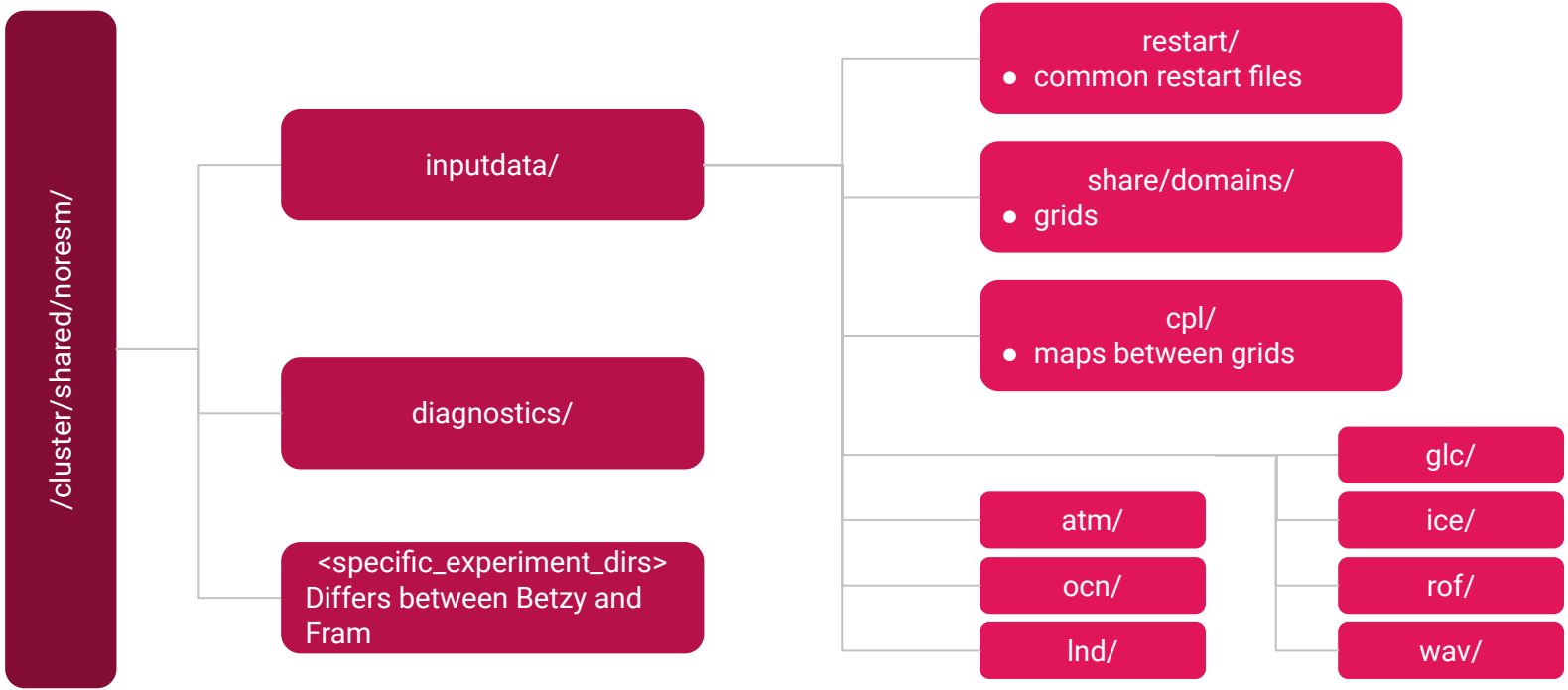
NorESM copies build and run files to subdirectories in $USERWORK. These files are not backed up,and are subject to automatic deletion after 42 days. Therefore, make sure you copy output files you want to keep to a permanent storage area.

Documentation: https://documentation.sigma2.no/files_storage/clusters.html

NorESM shared resources on Fram and Betzy

Shared resources for NorESM are available on both Fram and Betzy under directory:

`/cluster/shared/noresm/`



Where to learn more about HPC computing

- Sigma2 “getting started” web pages:
  https://documentation.sigma2.no/getting_started/getting_started.html

- Sigma2 “Running jobs” web page:
  https://documentation.sigma2.no/jobs/overview.html

- Sigma2/NRIS training material:
  https://documentation.sigma2.no/training/material.html

- NRIS training events:
  https://www.sigma2.no/training

Setting up a new NorESM case (Tomas)

- Creating a new case or cloning existing case
- Compsets
- Grids

Creating a new case

A new case is created by running the script `<NorESM>/cime/scripts/create_newcase` , where `<NorESM>` refers to the base directory of your personal clone of NorESM.

General command structure:

```
./create_newcase
        --case <path/to/casedir/casename>[required]
        --compset <compset_name>                    [required]
        --res <grid_name>                           [required]
        --machine <machine_name>                    [usually required (options: fram, betzy)]
        --project <project_name>                    [usually required (options: nn????k)]
        --pecount <label/number>                    [optional (number of cores required for job)]
        --user-mod-dirs <path/to/usermods>          [optional (used for some common run setups)]
        --run-unsupported                           [optional (used for some non-standard setups)]
```

Documentation: https://noresm-docs.readthedocs.io/en/latest/configurations/experiments.html

Creating a clone case

A clone case is created by running the script `<NorESM>/cime/scripts/create_clone` .

General command structure:

```
./create_clone
        --case <path/to/casedir/casename>[required]
        --clone <path/to/existing/case>                 [required]
        --project <project_name>                        [usually required (options: nn????k)]
        --user-mod-dirs <path/to/usermods>              [optional (used for some common run setups)]
        --keepexe                                       [optional (set EXEROOT link to original build)]
```

Cloning without `--keepexe` will copy all case files, but the case needs to be re-built (run `./case.build`) before submitting a job.

Cloning a case with `--keepexe` will not create a bld/ directory (no re-build required), and the SourceMods directory will be changed to a symbolic link pointing to the original case directory.

Compsets

- An experiment with some sets of components and forcing

- List of all compsets

`./query_config --compsets`	list all existing compsets;
`./query_config --compsets allactive`	all fully coupled compsets;
`./query_config --compsets blom`	all ocean-only compsets;
`./query_config --compsets cam`	all atmosphere-only compsets;

- All compsets starting with N are NorESM related compsets

Compset string

- The compset longname has the specified order atm, lnd, ice, ocn, river, glc, wave, cesm-options
- The notation for the compset longname is

TIME_ATM[%phys]_**LND**[%phys]_**ICE**[%phys]_**OCN**[%phys]_**ROF**[%phys]_**GLC**[%phys]_**WAV**[%phys][_ESP%phys][_BGC%phys]

TIME = Time period (e.g. 1850, 2000, HIST, SSP126, SSP245, SSP370. SSP585)
ATM = [CAM40, CAM50, CAM54, CAM60] ; LND = [CLM45, CLM50, SLND] ; ICE = [CICE, DICE, SICE]
OCN = [DOCN, ,AQUAP, SOCN, BLOM]; ROF = [RTM, MOSART, SROF] ; GLC = [CISM1, CISM2, SGLC]
WAV = [WW3, DWAV, XWAV, SWAV]; ESP = [SESP] ; BGC = optional BGC scenario

- The OPTIONAL %phys attributes specify submodes of the given system
- For example DOCN%DOM is the data ocean model for DOCN
- ALL the possible %phys choices for each component are listed.
- ALL data models must have a %phys option that corresponds to the data model mode
- Each compset node is associated with the following elements
- - lname ; alias ; support (optional description of the support level for this compset)
- Each compset node can also have the following attributes
- - grid (optional regular expression match for grid to work with the compset)

Documentation: https://noresm-docs.readthedocs.io/en/latest/configurations/experiments.html , https://noresm-docs.readthedocs.io/en/latest/configurations/cmip6_compsets.html#cmip6-deck-compsets

Some fully coupled compsets

<short_name> : (description)
TIME_ATM[%phys]_LND[%phys]_ICE[%phys]_OCN[%phys]_ROF[%phys]_GLC[%phys]_WAV[%phys][_ESP%phys][_BGC%phys]

N1850frc2 : (piControl)
1850_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS

NHISTfrc2 : (historical)
HIST_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS

NSSP126frc2 : (scenario)
SSP126_CAM60%NORESM%FRC2_CLM50%BGC-CROP_CICE%NORESM-CMIP6_BLOM%ECO_MOSART_SGLC_SWAV_BGC%BDRDDMS

Documentation: https://noresm-docs.readthedocs.io/en/latest/configurations/cmip6_compsets.html#cmip6-deck-compsets , https://noresm-docs.readthedocs.io/en/latest/configurations/cmip6_compsets.html#cmip6-scenario-compsets-only-frc2-compsets https://noresm-docs.readthedocs.io/en/latest/configurations/cmip6_compsets.html#reproduce-cmip6-picontrol-historical-and-ssp5-8-5-experiments

Where to find compset definitions

- Fully coupled compsets:

  `cime_config/config_compsets.xml`

- Component specific components (not fully coupled – data atmosphere, stub atmosphere, stub or data ocean, ..)

  `cam/cime_config/config_compsets.xml`
  `blom/cime_config/config_compsets.xml`
  `clm/cime_config/config_compsets.xml`
  `cice/cime_config/config_compsets.xml`
  `cism/cime_config/config_compsets.xml`

Grid resolution

Grid definition: `cime/config/cesm/config_grids.xml`

Grids for scientifically supported NorESM2 experiments include:

Atmosphere & land:
f19 : 1.9x2.5 = 1.9 degree latitude and 2.5 degree longitude resolution = 144x96
f09 : 0.9x1.25 = 0.9 degree latitude and 1.25 degree longitude resolution = 288x192

Ocean & sea ice:
tnx1v4 : tripolar 1 degree grid = 360x384

Data atmosphere: (OMIP experiments)
T62, **TL319** : approx. 2 degree Gaussian and 0.5 degree spectral grid

More grid configurations are defined, but they may not be available on a specific HPC or may require additions to NorESM2 config files in order to work properly.

Mapping files

- There is different resolution for ocean, atmosphere and river runoff grid
- Coupler is used to interpolate/transfer fields from one grid to another grid
- Map files are generated for weight factors (cime/tools) and these factors are stored; for that purpose ESMF tool is used.
- List of some map files: `cime/config/cesm/config_grids.xml`

  `map_tnx1v4_to_fv0.9x1.25_aave_da_170609.nc` (ocean → atmosphere)
  `map_tnx1v4_to_fv1.9x2.5_aave_da_170609.nc` (ocean → atmosphere)
  `map_fv0.9x1.25_to_tnx1v4_aave_da_170609.nc` (atmosphere → ocean)
  `map_fv1.9x2.5_to_tnx1v4_aave_da_170609.nc` (atmosphere → ocean)
  `map_r05_to_tnx1v4_e1000r300_170609.nc` (river → ocean)

Hands-on session 2 : checkout_externals

In release-noresm2.0.5 there is a second externals file:
Externals_continuous_development.cfg

This file is used to build NorESM with alternative development branches of model components.

1. **Look at differences between externals files:**
   diff -u Externals.cfg Externals_continuous_development.cfg
2. **Examine status for current and alternative externals file**
   ./manage_externals/checkout_externals -S ← **for current externals**
   -e Externals_continuous_development.cfg -S ← **for alternative externals**

Symbols used by status checker:

: all is fine	s : out-of-sync	o : optional source
e : empty	? : unknown (no .git or .svn)	M : modified source

Hands-on session 2 : query_config

Use query_config to learn more about NorESM model options

1. Go to scripts directory: `cd cime/scripts`
2. Find details about compsets N1850frc, NorESM coupled, and blom
   `./query_config --compsets | grep N1850frc2`
   `./query_config --compsets | awk '$1 ~ /^N/'`
   `./query_config --compsets blom`
3. Find config options for a component or data replacement
   `./query_config --components blom` ← Active ocean component
   `./query_config --components docn` ← Data ocean component
4. Find details about grids
   `./query_config --grids | grep -A 2 tn14` ← All grids using tn1v4 ocean grid

Configure build and run settings for a case (Ada)

- env_mach_pes.xml
- user_nl_*
- env_run.xml
- env_batch.xml

Basic steps to run NorESM: Set up the case

```
$ ./xmlchange NTASKS_OCN=123

$ vi env_mach_pes.xml
```

During this workshop we will run NorESM on 1 node.

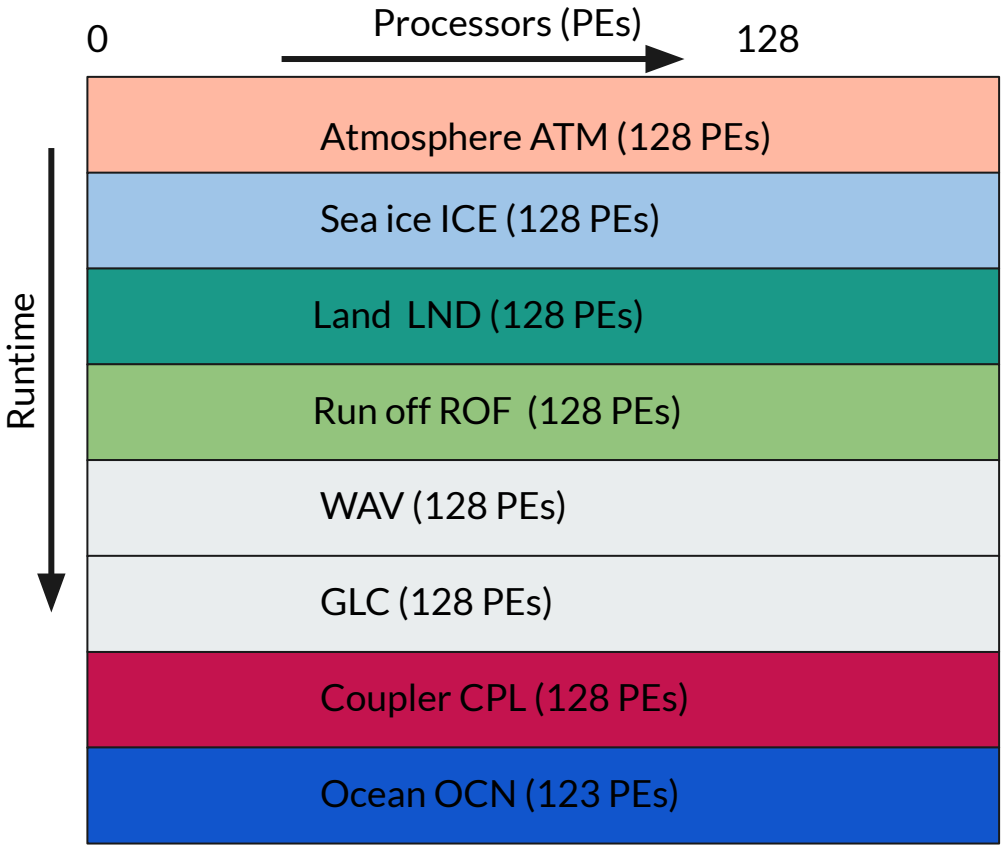
Change to 123 on line 63 (you can type 63G to get to the correct line) and type i (insert) to make changes:

```
<entry id="NTASKS">
  <type>integer</type>
  <values>
   <value compclass="ATM">128</value>
   <value compclass="CPL">128</value>
   <value compclass="OCN">128</value>
   <value compclass="WAV">128</value>
   <value compclass="GLC">128</value>
   <value compclass="ICE">128</value>
   <value compclass="ROF">128</value>
   <value compclass="LND">128</value>
   <value compclass="ESP">1</value>
  </values>
  <desc>number of tasks for each component</desc>
 </entry>
```

```
<entry id="NTASKS">
  <type>integer</type>
  <values>
   <value compclass="ATM">128</value>
   <value compclass="CPL">128</value>
   <value compclass="OCN">123</value>
   <value compclass="WAV">128</value>
   <value compclass="GLC">128</value>
   <value compclass="ICE">128</value>
   <value compclass="ROF">128</value>
   <value compclass="LND">128</value>
   <value compclass="ESP">1</value>
  </values>
  <desc>number of tasks for each component</desc>
 </entry>
```

```
$ ./case.setup
```

Configure run settings: env_mach_pes.xml

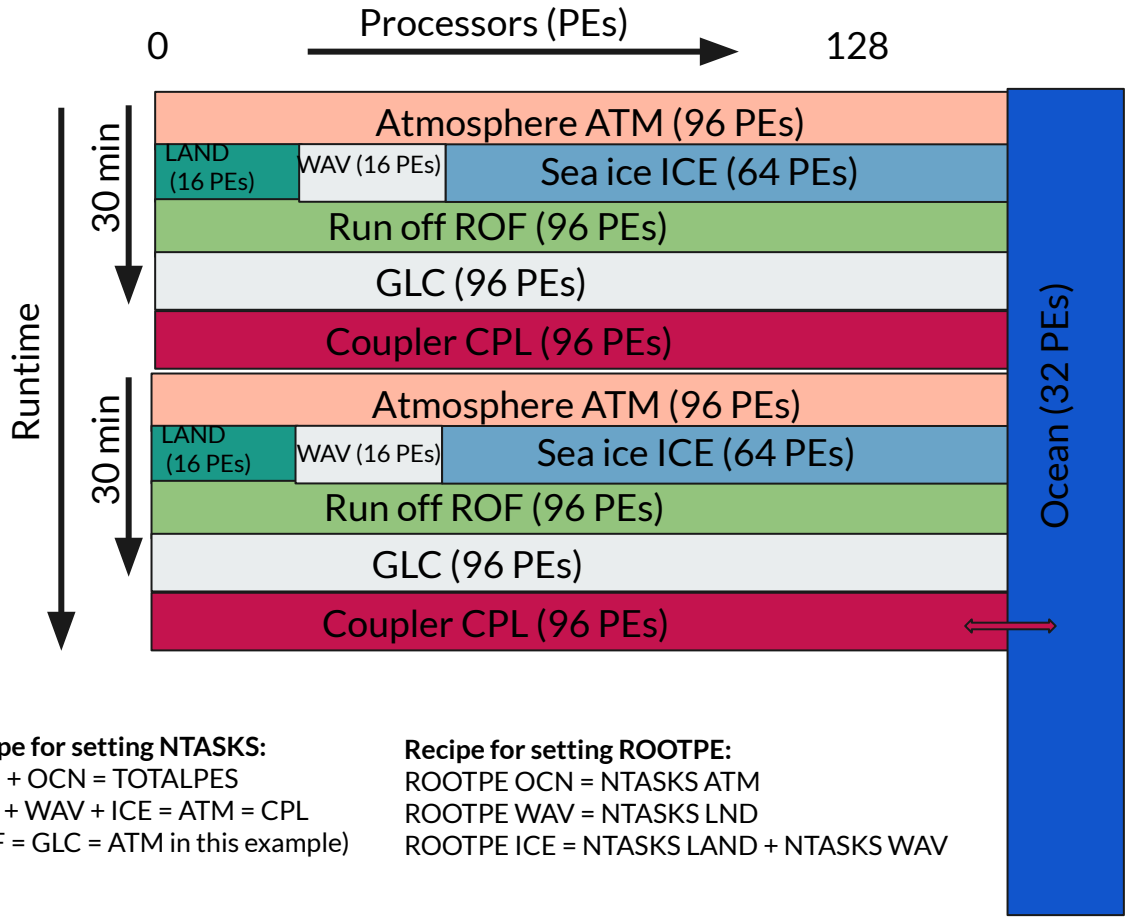


In the previous case we used 128 PEs with each component running sequentially over the entire set of processors:

```
<entry id="COST_PES" value="128">
    <type>integer</type>
    <desc>pes or cores used relative to MAX_MPITASKS_PER_NODE for accounting (0
means TOTALPES is valid)</desc>
  </entry>
  <entry id="TOTALPES" value="128">
    <type>integer</type>
    <desc>total number of physical cores used (setup automatically - DO NOT
EDIT)</desc>
  </entry>

 <entry id="NTASKS">
    <type>integer</type>
    <values>
     <value compclass="ATM">128</value>
     <value compclass="CPL">128</value>
     <value compclass="OCN">123</value>
     <value compclass="WAV">128</value>
     <value compclass="GLC">128</value>
     <value compclass="ICE">128</value>
     <value compclass="ROF">128</value>
     <value compclass="LND">128</value>
     <value compclass="ESP">1</value>
    </values>
    <desc>number of tasks for each component</desc>
  </entry>
```

WAV and GLC are stub components which are present only to meet interface requirements, but not active model components in NorESM. Still need to set NTASKS!

Configure run settings: env_mach_pes.xml



Recipe for setting NTASKS:
ATM + OCN = TOTALPES
LND + WAV + ICE = ATM = CPL
(ROF = GLC = ATM in this example)

Recipe for setting ROOTPE:
ROOTPE OCN = NTASKS ATM
ROOTPE WAV = NTASKS LND
ROOTPE ICE = NTASKS LAND + NTASKS WAV

Now, we still use 128 PEs, running on 1 node:

```
<entry id="NTASKS">
   <type>integer</type>
   <values>
    <value compclass="ATM">96</value>
    <value compclass="CPL">96</value>
    <value compclass="OCN">32</value>
    <value compclass="WAV">16</value>
    <value compclass="GLC">96</value>
    <value compclass="ICE">64</value>
    <value compclass="ROF">96</value>
    <value compclass="LND">16</value>
    <value compclass="ESP">1</value>
   </values>
   <desc>number of tasks for each component</desc>
  </entry>
```

PE “Starting point”

```
<entry id="ROOTPE">
   <type>integer</type>
   <values>
    <value compclass="ATM">0</value>
    <value compclass="CPL">0</value>
    <value compclass="OCN">96</value>
    <value compclass="WAV">16</value>
    <value compclass="GLC">0</value>
    <value compclass="ICE">32</value>
    <value compclass="ROF">0</value>
    <value compclass="LND">0</value>
    <value compclass="ESP">0</value>
   </values>
   <desc>ROOTPE (mpi task in MPI_COMM_WORLD) for each
component</desc>
  </entry>
```

Configure run settings: env_mach_pes.xml

Usually we run on more nodes. When building a case on BETZY you can set the number of nodes by setting --pecount = S, M, L or X1 when creating a case. Then you don't need to change env_mach_pes.xml:

- NorESM2-LM (grid = f19_tn14)
  - S = 4 nodes
  - M = 8 nodes
  - X1 = 10 nodes

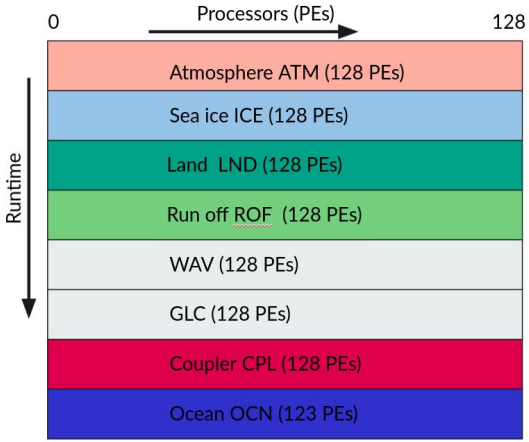
- NorESM2-MM (grid = f09_tn14)
  - S = 4 nodes
  - M = 9 nodes
  - L = 15 nodes
  - X1 = 17 nodes

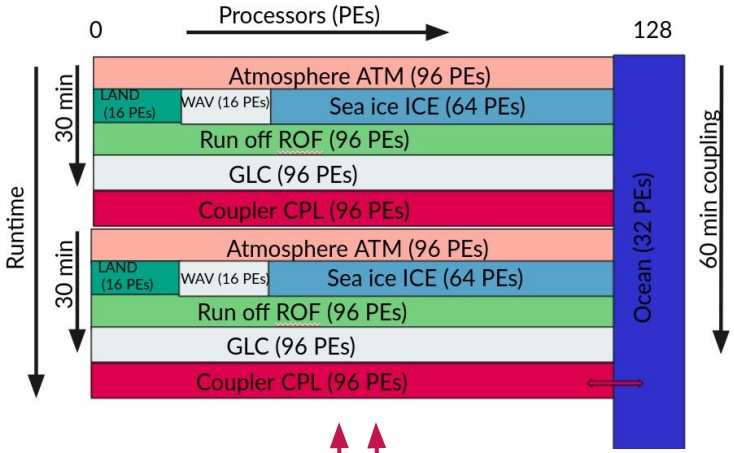
E.g. for running NorESM2-LM on 8 nodes:

```
$ cd ~/NorESM/noresm2.0/cime/scripts/

$ ./create_newcase --case ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115 --mach betzy --res
f19_tn14 --compset N1850frc2 --project nn9560k --user-mods-dir cmip6_noresm_DECK --pecount=S
```

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/platforms.html#setting-number-of-nodes-on-betzy

Configure run settings: why do we care?





Cost:

- 1 node (--pecount = 128):
  - Model Cost: 729.36 pe-hrs/simulated_year
  - Model Throughput: 4.21 simulated_years/day
  - TOT Run Time: 1742.223 seconds
- 2 noder (--pecount = 256):
  - Model Cost: 802.65 pe-hrs/simulated_year
  - Model Throughput: 7.65 simulated_years/day
  - TOT Run Time: 958.643 seconds
- 4 noder (--pecount = 512):
  - Model Cost: 1010.31 pe-hrs/simulated_year
  - Model Throughput: 12.16 simulated_years/day
  - TOT Run Time: 603.331 seconds
- 8 noder (--pecount = 1024):
  - Model Cost: 1634.02 pe-hrs/simulated_year
  - Model Throughput: 15.04 simulated_years/day
  - TOT Run Time: 487.899 seconds

Cost:

- 1 node (--pecount =):
  - Model Cost: 865.16 pe-hrs/simulated_year
  - Model Throughput: 3.55 simulated_years/day
  - TOT Run Time: 2066.618 seconds
- 2 noder (--pecount = 2x):
  - Model Cost: 932.70 pe-hrs/simulated_year
  - Model Throughput: 6.59 simulated_years/day
  - TOT Run Time: 1113.970 seconds
- 4 noder (--pecount = S):
  - Model Cost: 924.545 pe-hrs/simulated_year
  - Model Throughput: 13.29 simulated_years/day
  - TOT Run Time: 552.269 seconds
- 8 noder (--pecount = M):
  - Model Cost: 1241.76 pe-hrs/simulated_year
  - Model Throughput: 19.79 simulated_years/day
  - TOT Run Time: 370.774 seconds

Configure run settings: env_mach_pes.xml

```
[adagj@login-1.BETZY ~]$ cd NorESM/cases/N1850frc2_f19_tn14_test02_20211115/
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_ATM=96
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_CPL=96
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_GLC=96
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_ROF=96
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_OCN=32
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_WAV=16
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_LND=16
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_ESP=1
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange NTASKS_ICE=64
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange ROOTPE_ICE=32
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange ROOTPE_OCN=96
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./xmlchange ROOTPE_WAV=16
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$ ./case.setup
Setting resource.RLIMIT_STACK to -1 from (8388608, -1)
/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_20211115/env_mach_specific.xml already exists, delete to replace
job is case.run USER_REQUESTED_WALLTIME None USER_REQUESTED_QUEUE None
Creating batch scripts
Writing case.run script from input template /cluster/home/adagj/NorESM/noresm2.0/cime/config/cesm/machines/template.case.run
Creating file .case.run
Writing case.st_archive script from input template /cluster/home/adagj/NorESM/noresm2.0/cime/config/cesm/machines/template.st_arc
hive
Creating file case.st_archive
Creating user_nl_xxx files for components and cpl
If an old case build already exists, might want to run 'case.build --clean' before building
You can now run './preview_run' to get more info on how your case will be run
[adagj@login-1.BETZY ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115]$
```

Basic steps to run NorESM: Set up the case

Enter the case folder:

```
$ cd ~/NorESM/cases/N1850frc2_f19_tn14_test01_20211115/
```

./case_setup : Creates namelists and various files and directories needed to build and run the case.

- Any changes to env_mach_pes.xml and env_case.xml must be made before running ./case.setup
- To run this initially for the experiment, simply run: ./case.setup
- To rerun after making changes to env_mach_pes.xml, run: ./case.setup --reset
- ./case.setup --clean do not remove user_nl_* files and Macros file

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/newbie-guide.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiments.html#create-and-configure-a-new-case
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#machine-specific-environment

Configure run setting: user namelists

In user_nl_$component you can set e.g. :

- new parameter values
- restart files
- output variables and time frequency
- activate e.g. amount of aerosol output

E.g. to change only the initial state of the sea ice, in user_nl_cice:

```
&setup_nml
 ice_ic = "/cluster/work/users/adagj/noresm/N1850frc2_f19_tn14_keyclim_snow/run/N1850_piControl_snow_KeyClim.cice.r.1855-01-01-00000.nc"
```

E.g. change some (mixing) parameters in BLOM, in user_nl_blom:

```
EGC     = 2.0
EGIDFQ   = 0.25
BDMC2 = .15
NIWGF = .5
```

Unique for user_nl_blom: it does not matter which namelist group the variable belongs to

E.g. to double the atmospheric CO2 concentration, in user_nl_cam:

```
&chem_surfvals_nl
 co2vmr        =    568.64e-6
```

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#user-namelists
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#user-namelists
- https://noresm-docs.readthedocs.io/en/latest/output/aerosol_output.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/omips.html#modify-user-namelist-for-blom-ihamocc
- https://noresm-docs.readthedocs.io/en/latest/configurations/omips.html#modify-user-name-lists-for-cice
- https://noresm-docs.readthedocs.io/en/latest/configurations/clm.html#user-name-list-modifications

Configure run setting: SourceMods

The SourceMods folder: for **code modification(s)** beyond what is possible from user namelists

- contains sub-directories for all model component.
- Make a copy of the fortran file(s) you want to modify in the relevant sub-folder and modify the file(s) as needed before building the model.
- When compiling, the model will prioritize the modified file located under the SourceMods folder over the default version of the file located in the model source code under <noresm-base>.
- **Aerosol diagnostics and output** can be enabled by the use of SourceMods:
  - <case_folder>/**SourceMods**/src.cam/preprocessorDefinitions.h
  - #define AEROCOM: additionally 149 variables are written (+ ca. 13% CPU-time)
  - #define AEROFFL: additional radiation-diagnostics for aerosol indirect effect. Gives 8 additionally variables (+ ca. 5% CPU-time).

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html?highlight=SourceMods#code-modifications
https://noresm-docs.readthedocs.io/en/latest/output/aerosol_output.html?highlight=SourceMods#decomposition-of-aerosol-direct-semidirect-and-indirect-radiative-forcing

Configure run setting: --user_mods_dir

-- user-mods-dir is important for the *output*, and should be changed according to your needs.

The usermods under noresm2.0/cime_config/usermods_dirs/ include:

cmip6_noresm_DECK (AEROFFL)
cmip6_noresm_hifreq (high frequency output, AEROFFL)
cmip6_noresm_hifreq_xaer (high frequency output, AEROFFL **and** AEROCOM)
cmip6_noresm_keyCLIM (used **for** KeyCLIM experiments, AEROFFL)
cmip6_noresm_xaer (AEROFFLand AEROCOM)

To activate the cmip6_noresm_DECK usermod, run the create_newcase script with the option --user-mods-dir cmip6_noresm_DECK

Remember that the amount of diagnostics and the output frequency have a huge impact on both the run time and storage.

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html
https://noresm-docs.readthedocs.io/en/latest/output/standard_output.html
https://noresm-docs.readthedocs.io/en/latest/output/aerosol_output.html

Configure run setting: user namelists

env_mach_pes.xml and user_nl_component must be changed **before building**

In the case directory, there is a **CaseDocs** folder:

- there you'll find the namelists (i.e. component_in) containing all the input files and parameters used.
- e.g. CaseDocs/ocn_in , CaseDocs/atm_in , CaseDocs/lnd_in etc.
- *You should never have to edit the contents of this directory.*
- If you wish to make changes to the component_in files, you need to change user_nl_component and rebuild the case.

```
$ ./case.build
```

env_run.xml and env_batch.xml can be changed **after building**

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html

Basic steps to run NorESM: Building the case

Creating an executable

```
$ ./case.build
```

- You can see all software modules in **env_mach_specific.xml** and all compiler flags in **Macros.make**
- Processors configuration in **env_mach_pes.xml**
- ./case.build – it will create namelist files and compile all the required libraries (mct, gptl, csm_share and pio) and components (cam, blom, clm, cice, ..).
- Finally, build the binary **cesm.exe**; found in /cluster/work/users/$USER/noresm/$CASENAME/bld

After this, you can modify only **env_batch.xml** and **env_run.xml**:

BUT…. if you change your mind, make a clean:
./case.build --clean component removes object files of components
or
./case.build --clean-all removes bld directory
and try again:
./case.build

Configure run settings: env_run.xml

Branch run

In a branch run, all components are initialized using a consistent set of restart files from a previous run. Mostly used for tuning experiments and investigating parameter space

- RUN_TYPE to “branch”
- RUN_REFDIR directory containing reference data
- RUN_REFCASE name of reference case
- RUN_REFDATE Reference date branch run
- GET_REFCASE TRUE else you have to copy data to run folder

Hybrid run

Not as strict as a branch run, all components are initialized but can have reference files from several experiments. Used e.g. for a historical experiment starting from piControl

- RUN_TYPE to “hybrid”
- RUN_REFDIR directory containing reference data
- RUN_REFCASE name of reference case
- RUN_REFDATE Reference date branch run
- GET_REFCASE TRUE else you have to copy data to run folder
- RUN_STARTDATE set the date for the beginning of the simulation

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#some-common-configuration-settings

Configure run settings: env_run.xml

Setting the length of the simulation:

- STOP_OPTION : nseconds,nsecond,nminutes,nhours,nhour,ndays,nmonths, nyears
- STOP_N: numerical value

Writing restart files in middle of simulation:
Restart files are written end of the simulation by default; But, if you are having a long simulation of 100 years; for safety reason you want to write restart files at some frequency you can set below option

- REST_OPTION: nseconds,nsecond,nminutes,nminute,nhours,nhour,nmonths, nyears
- REST_N: numerical value
- DOUT_S_SAVE_INTERIM_RESTART_FILES: TRUE or FALSE. Set to TRUE if you want to archive all restart files and FALSE if you only want to archive restart files from the end of the simulation

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#some-common-configuration-settings

Configure run settings: env_run.xml

Continue a simulation: when you are having WALLCLOCK time limitation on system. For example, you want to have 200 years simulation and WALLCLOCK time limitation is 5 days; you are able to simulate 10 model years/day; to complete 200 model years simulation set RESUBMIT=3, STOP_N to 50 and STOP_OPTION to nyears

- **CONTINUE_RUN** in env_run.xml; TRUE or FALSE.
  - You need all restart files and rpointer.* files in run folder.
  - Please note that CONTINUE_RUN needs to be FALSE first time you submit an experiment.
  - Will automatically be set to TRUE if the job is automatically resubmitted, i.e. if RESUBMIT > 0

- **RESUBMIT** in env_run.xml ; an integer value.
  - will auto resubmit till specified value; you will have total simulation period STOP_N*(RESUBMIT+1)

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#some-common-configuration-settings

Configure run settings: env_run.xml

Two scripts which you may (or may not) find helpful (located in the case folder):

1. xmlquery: provides the information and its value which are set in the *.xml files, e.g. env_run.xml
2. xmlchange: used to change values/parameters set in the *.xml files, e.g. in env_run.xml

```
$ ./xmlquery --value STOP_OPTION,STOP_N
ndays,5
$ ./xmlchange STOP_OPTION=nyears,STOP_N=1
$ ./xmlquery --value STOP_OPTION
nyears,1
```

I usually just open env_run.xml in vim and change whatever I need to change:

```
$ vi env_run.xml
```

Documentation:

- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html
- https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#some-common-configuration-settings

Configure run setting: env_batch.xml

You also need to modify **env_batch.xml:**

- XML block for case.run
- env_batch.xml sets the arguments for the batch job commands
- Neet to set JOB_WALLCLOCK_TIME
- XML block for case.st_archive
- case.st_archive is a pending job which moves files from the run directory to the archive directory after a successful simulation.
- also here you need to set JOB_WALLCLOCK_TIME
- you can also modify project for CPU hours if required
  (usually it is set during experiment creation, but you may need to change it)

After setting the walltime for the two jobs, you can submit your case:

```
$ ./case.submit
```

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/experiment_environment.html#batch-job-environment

Experiment status and timing statistics

In your case directory: the file CaseStatus logs all information on what you have done and if it worked or not:

```
2021-11-10 14:45:01: case.setup starting
---------------------------------------------------
2021-11-10 14:45:02: case.setup success
---------------------------------------------------
2021-11-10 14:45:07: case.build starting
---------------------------------------------------
2021-11-10 14:51:31: case.build success
---------------------------------------------------
2021-11-10 14:52:04: case.submit starting
---------------------------------------------------
2021-11-10 14:52:12: case.submit error
ERROR: Command: 'sbatch --time 00:59:00 -q devel --account nn9560k .case.run --resubmit' failed with error 'sbatch: error: QOSGrpNodeLimit
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)' from dir '/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_2021
1115'
---------------------------------------------------
2021-11-10 14:52:52: case.submit starting
---------------------------------------------------
2021-11-10 14:52:59: case.submit error
ERROR: Command: 'sbatch --time 00:29:00 -q devel --account nn9560k .case.run --resubmit' failed with error 'sbatch: error: QOSGrpNodeLimit
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)' from dir '/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_2021
1115'
---------------------------------------------------
2021-11-10 14:53:32: case.submit starting
---------------------------------------------------
2021-11-10 14:53:39: case.submit error
ERROR: Command: 'sbatch --time 00:29:00 -q devel --account nn9560k .case.run --resubmit' failed with error 'sbatch: error: QOSGrpNodeLimit
sbatch: error: Batch job submission failed: Job violates accounting/QOS policy (job submit limit, user's size and/or time limits)' from dir '/cluster/home/adagj/NorESM/cases/N1850frc2_f19_tn14_test02_2021
1115'
---------------------------------------------------
2021-11-10 14:57:08: case.setup starting
---------------------------------------------------
2021-11-10 14:57:09: build.clean starting
---------------------------------------------------
2021-11-10 14:57:27: build.clean success
---------------------------------------------------
2021-11-10 14:57:28: case.setup success
---------------------------------------------------
2021-11-10 14:57:52: case.build starting
---------------------------------------------------
2021-11-10 14:59:37: case.build success
---------------------------------------------------
2021-11-10 15:00:26: case.submit starting
---------------------------------------------------
2021-11-10 15:00:34: case.submit success case.run:253485, case.st_archive:253486
---------------------------------------------------
2021-11-10 15:05:37: case.run starting
---------------------------------------------------
2021-11-10 15:05:46: model execution starting
---------------------------------------------------
2021-11-10 15:15:35: model execution success
---------------------------------------------------
2021-11-10 15:15:35: case.run success
---------------------------------------------------
2021-11-10 15:15:56: st_archive starting
---------------------------------------------------
2021-11-10 15:16:29: st_archive success
---------------------------------------------------
```

Experiment status and timing statistics

In your case directory: in the timing sub-directory; the file cesm_timing.$CASE provides information on grid type, run length, compset, processors configuration and many others.
Most important are timing statics : Model throughput, Model cost and run time

```
 Overall Metrics:
   Model Cost:             985.36   pe-hrs/simulated_year
   Model Throughput:         3.12   simulated_years/day

   Init Time   :     201.411 seconds
   Run Time    :     379.632 seconds       75.926 seconds/day
   Final Time  :       0.183 seconds

   Actual Ocn Init Wait Time     :       5.450 seconds
   Estimated Ocn Init Run Time   :       2.344 seconds
   Estimated Run Time Correction :       0.000 seconds
     (This correction has been applied to the ocean and total run times)

Runs Time in total seconds, seconds/model-day, and model-years/wall-day
CPL Run Time represents time in CPL pes alone, not including time associated with data exchange with other components

   TOT Run Time:     379.632 seconds       75.926 seconds/mday         3.12 myears/wday
   CPL Run Time:       8.458 seconds        1.692 seconds/mday       139.93 myears/wday
   ATM Run Time:     233.509 seconds       46.702 seconds/mday         5.07 myears/wday
   LND Run Time:     106.004 seconds       21.201 seconds/mday        11.17 myears/wday
   ICE Run Time:      36.725 seconds        7.345 seconds/mday        32.23 myears/wday
   OCN Run Time:     281.239 seconds       56.248 seconds/mday         4.21 myears/wday
   ROF Run Time:      18.111 seconds        3.622 seconds/mday        65.35 myears/wday
   GLC Run Time:       0.000 seconds        0.000 seconds/mday         0.00 myears/wday
   WAV Run Time:       0.000 seconds        0.000 seconds/mday         0.00 myears/wday
   ESP Run Time:       0.000 seconds        0.000 seconds/mday         0.00 myears/wday
   CPL COMM Time:    127.679 seconds       25.536 seconds/mday         9.27 myears/wday
```

Model Throughput increases to ~20 simulated_years/day if you use 4 nodes instead of 1

Experiment status and timing statistics

In your RUN directory: /cluster/work/users/$USER/noresm/$CASE/run/
You will find all the namelists; component_in files, the timing folder, restart files and rpointer.* and if your model simulation for some reason crashes; the log files.

- the cesm.log.$JOBID file can provide some hints of which component caused the crash
- the component.log.$JOBID file can provide information about which subroutine caused the crashed

If your model simulation is successful, the log files are archived in:
/cluster/work/users/$USER/archive/$CASE/logs/

For a successful simulation the cpl.log.$JOBID file will end with:

```
(seq_mct_drv): ===============          SUCCESSFUL TERMINATION OF CPL7-cesm ===============
(seq_mct_drv): ===============        at YMD,TOD =   00010106       0         ===============
(seq_mct_drv): ===============  # simulated days (this run) =        5.000  ===============
(seq_mct_drv): ===============  compute time (hrs)          =        0.105  ===============
(seq_mct_drv): ===============  # simulated years / cmp-day =        3.118  ===============
(seq_mct_drv): ===============  pes min memory highwater  (MB)      -0.001  ===============
(seq_mct_drv): ===============  pes max memory highwater  (MB)      -0.001  ===============
(seq_mct_drv): ===============  pes min memory last usage (MB)     521.612  ===============
(seq_mct_drv): ===============  pes max memory last usage (MB)    1174.614  ===============
```

If you find your experiment great: Create a clone

If you really like your set up and you want to run very similar cases, you can make a clone (or several) of your case. The clone will be set up as if it was created with the same create_newcase options as the existing case (except the case name) and will have identical env_*.xml, user_nml_<component> and SourceMods files (these files can of course be modified before building the case).

The **create_clone** script is an executable python script located in cime/scripts

./create_clone --case $PATH_TO_cases/$CASENAME --clone $PATH_TO_cases/$CLONENAME

--case $CASENAME -> name of the NorESM experiment you are creating
--clone $CLONENAME -> of the case you want to clone

$ cd ~/NorESM/noresm2.0/cime/scripts/

$ ./create_clone --case ~/NorESM/cases/N1850frc2_f19_tn14_test03_20211115 --clone ~/NorESM/cases/N1850frc2_f19_tn14_test02_20211115

./create_clone --help will provide you all input options including a description

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/experiments.html?highlight=create_clone#create-a-clone-case

Hands-on session 3

A) piControl (N1850)
- create a new case with resolution f19_tn14
- Set --pecount=S
- make it a branch run
- reference case: N1850_f19_tn14_11062019
- reference date: 1600-01-01
- 1 mnd simulation time

B) historical (NHIST)
- create a new case with resolution f19_tn14
- Set --pecount=S
- make it a hybrid run
- reference case: N1850_f19_tn14_11062019
- reference date: 1600-01-01
- 1 month + 1 month simulation time i.e. run for 1 month, resubmit simulation once and run for 1 more month (set RESUBMIT=1)
- Dump restart files every 10 days

Path to restart files:
/cluster/projects/nn9560k/userWorkShop_restfiles/NorESM2-LM/piControl/1600-01-01-00000/

Documentation:
https://noresm-docs.readthedocs.io/en/latest/configurations/cmip6_compsets.html#reproduce-cmip6-picontrol-historical-and-ssp5-8-5-experiments

Log files (Tomas)

Log files produced by NorESM

NorESM produce log output for case creation, build process, run process and archiving process.

Log files in the case directory:

CaseStatus : Summary of all processes (case setup, build, job submission and job completion).
Good place to start looking at the logs!

README.case : Log for case creation

software_environment.txt : System environments (modules) used during model build

logs/run_environment.txt.<id>.<time> : System environment (modules) loaded during model run.

<casename> : Log for job execution, CPU, memory and disk usage.

Documentation: https://noresm-docs.readthedocs.io/en/latest/output/noresm_logs.html

Log files produced by NorESM

Build logs : `<workdir>/noresm/<casename>/bld/` (<workdir> = /cluster/work/users/<username>)

`cesm.bldlog.<timestamp>.gz`	: Build the coupled model executable `cesm.exe`.
`<component>.bldlog.<timestamp>.gz`	: Build log for individual model components.

Run logs : `<workdir>/archive/<casename>/logs/` (assuming run and archive processes finished)

`cesm.log.<jobid>.<timestamp>.gz`	: Run log for coupled model system.
`<component>.log.<jobid>.<timestamp>.gz`	: Run log for individual model components.

If NorESM fails during run, you may find these log files in `<workdir>/noresm/<casename>/run/` .

Archiving logs : `<workdir>/archive/<casename>/`

`archive.log.<timestamp>`	: Log for short term archiving of model output.
`case.log`	: Log for archiving of case directory.

NOTE: Archived *.gz files should be expanded before reading: `gunzip <filename>.gz`

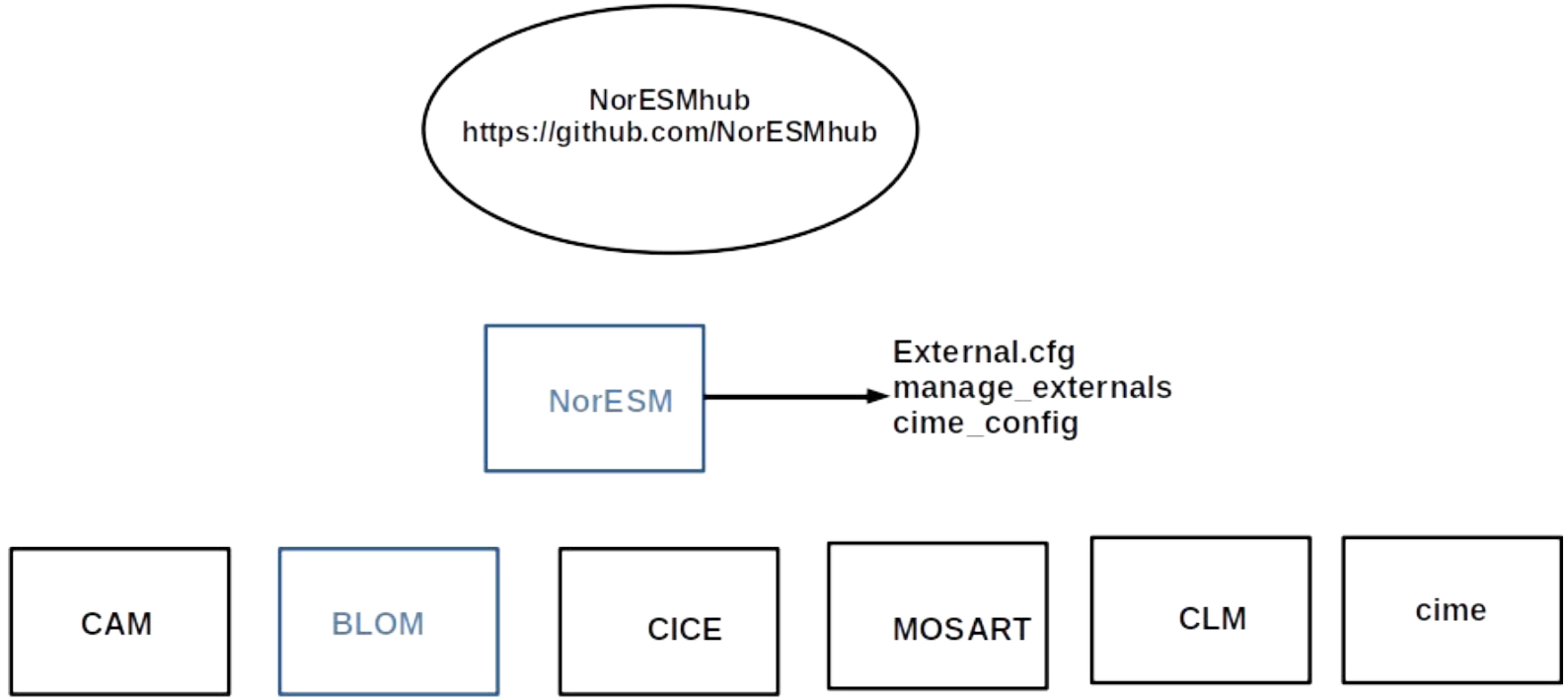
Include your own code changes (Tomas)

- Work with fork repository on github

Contribute your own code to NorESM: basic steps

1. Create a user account on gitHub https://docs.github.com/en/get-started/signing-up-for-github/signing-up-for-a-new-github-account
    - A basic account is sufficient. Note that all NorESM repositories are public by default.
2. Fork NorESM and/or a NorESM model component to your own account
3. Create an issue in the main repository describing the changes you want to make
    - Mainly used for discussions on severe bugs or feature changes, not so much for minor changes (e.g. documentation, small config changes)
4. Make changes in your fork repository
    - To minimize risk of code conflicts, do not make changes directly in branches that duplicate the main repository, but create “throw away” branches for your changes
5. Create a pull request to the main repository
6. Keep things in sync:
    - Remove your “throw away” branches after merging
    - Sync your fork with the main repository

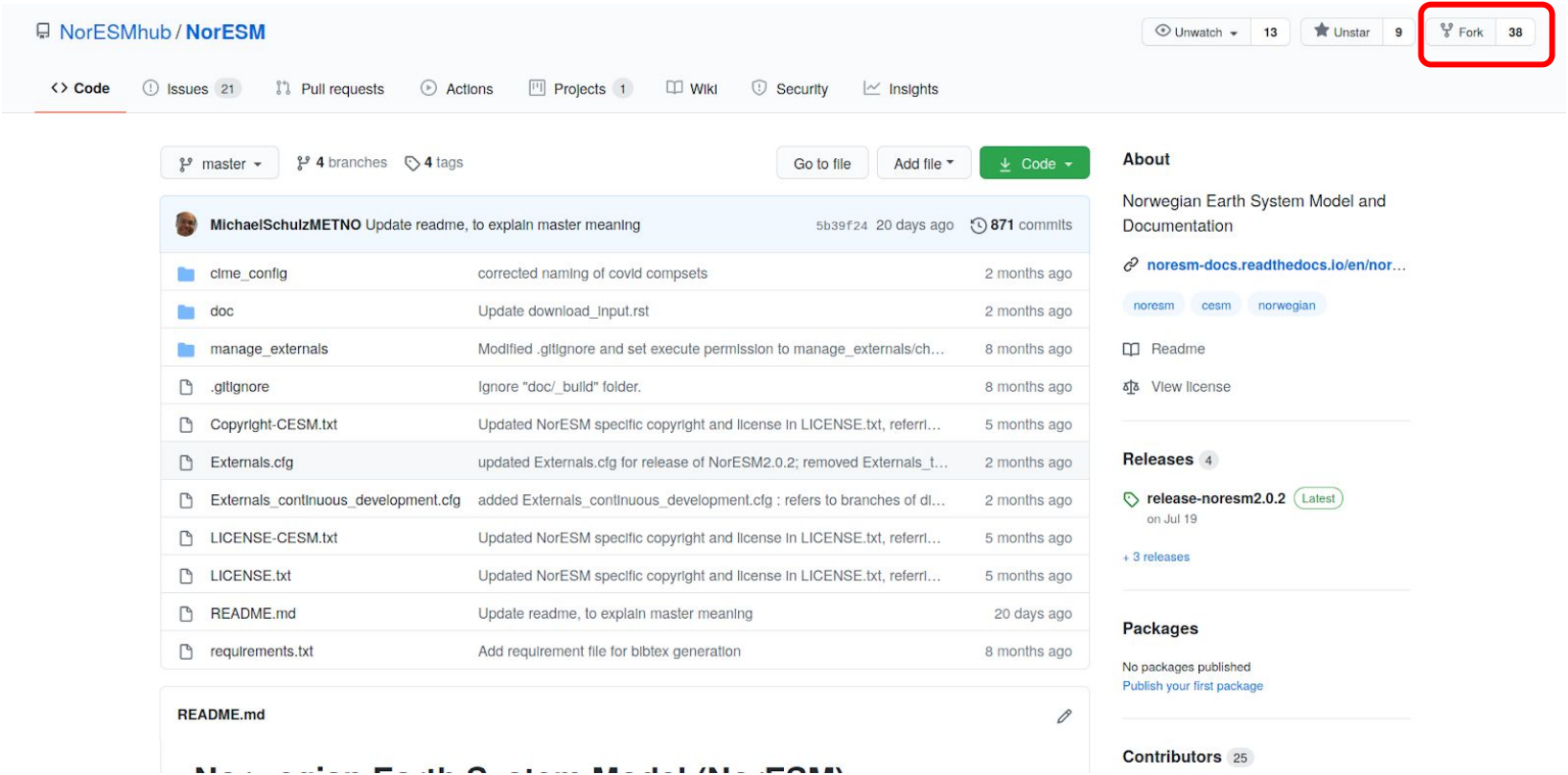
NorESM git repositories



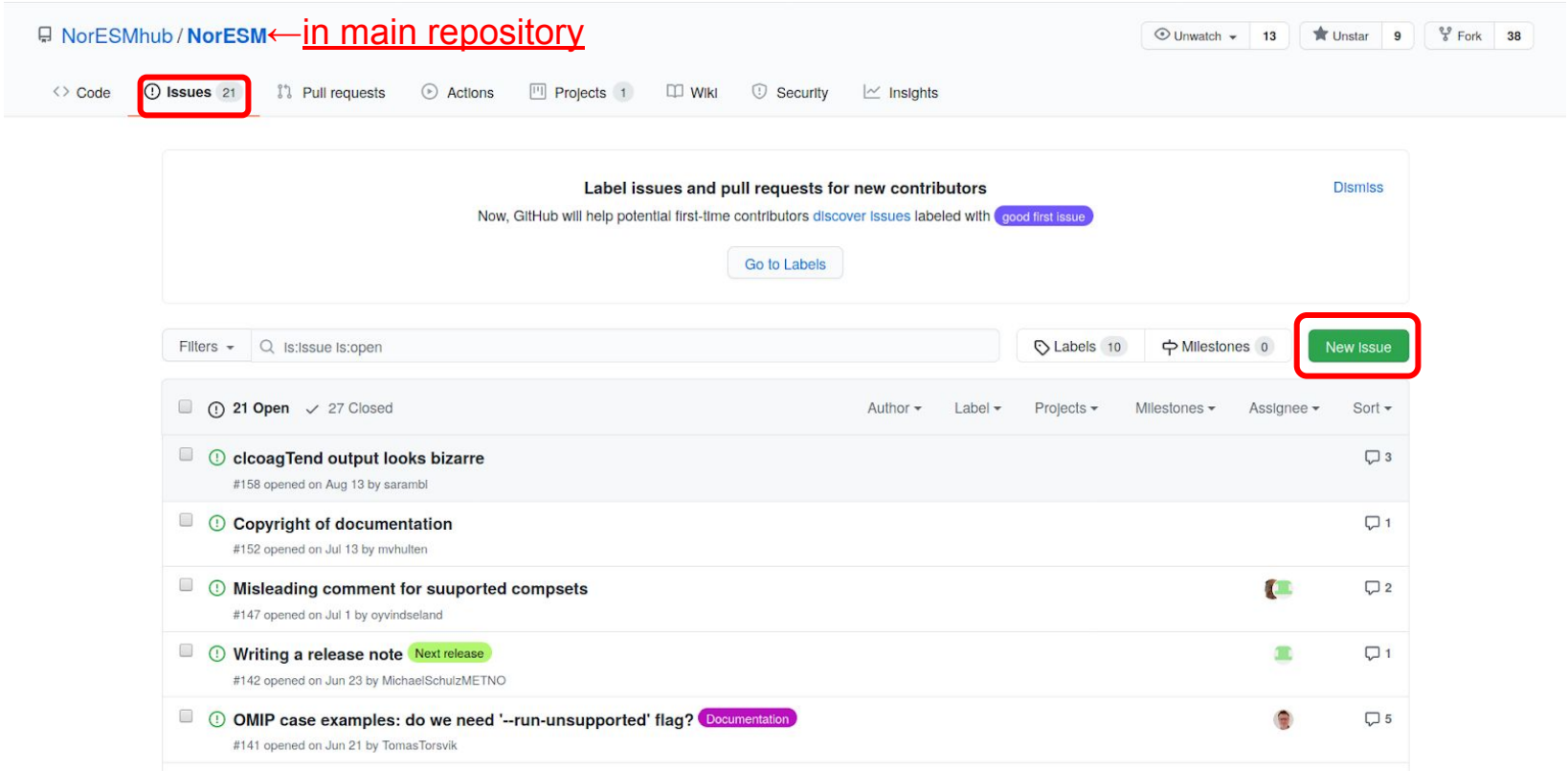
Repository in black are forked from CESM

Fork repository

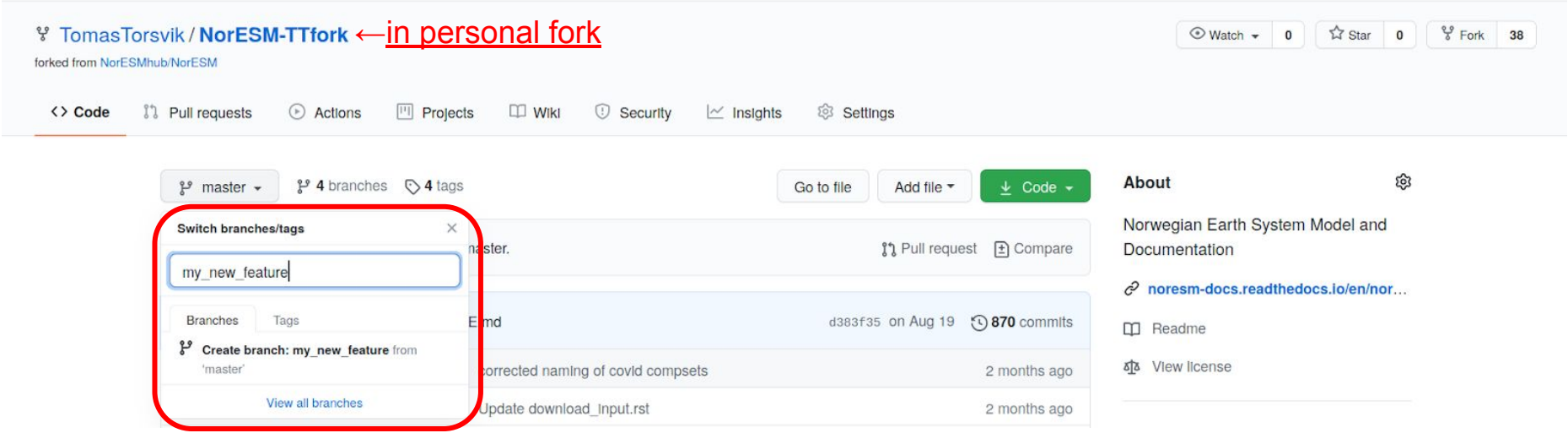
Fork NorESM from gitHub



Create an issue : discussion forum for code changes

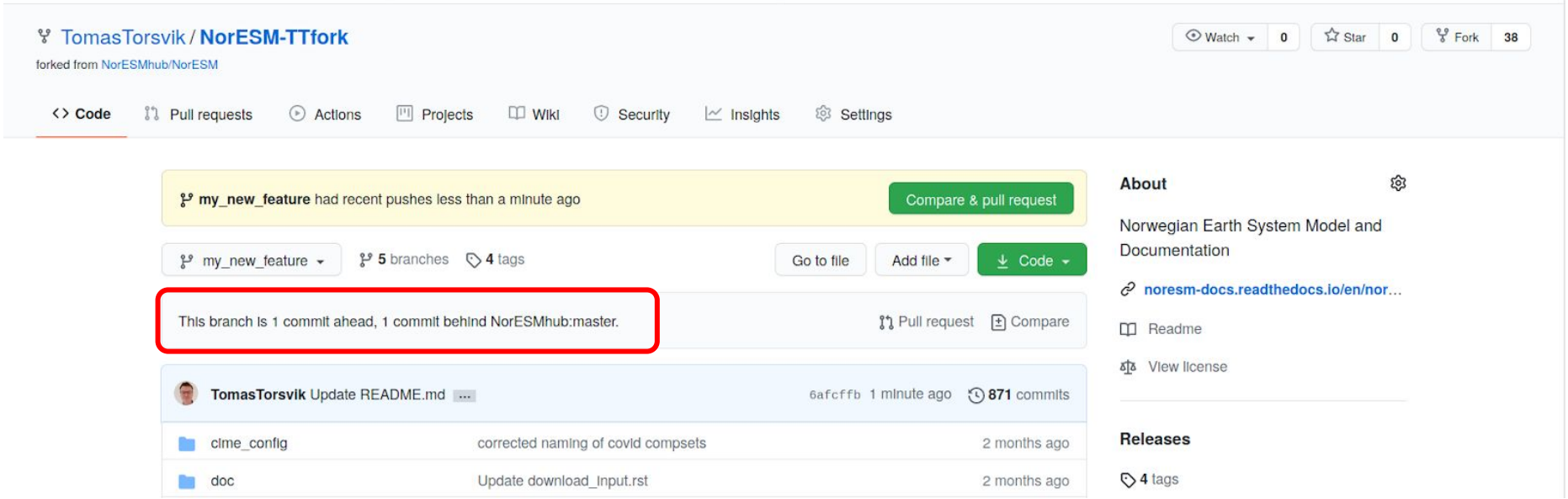


Create a fork : feature branch (highly recommended)

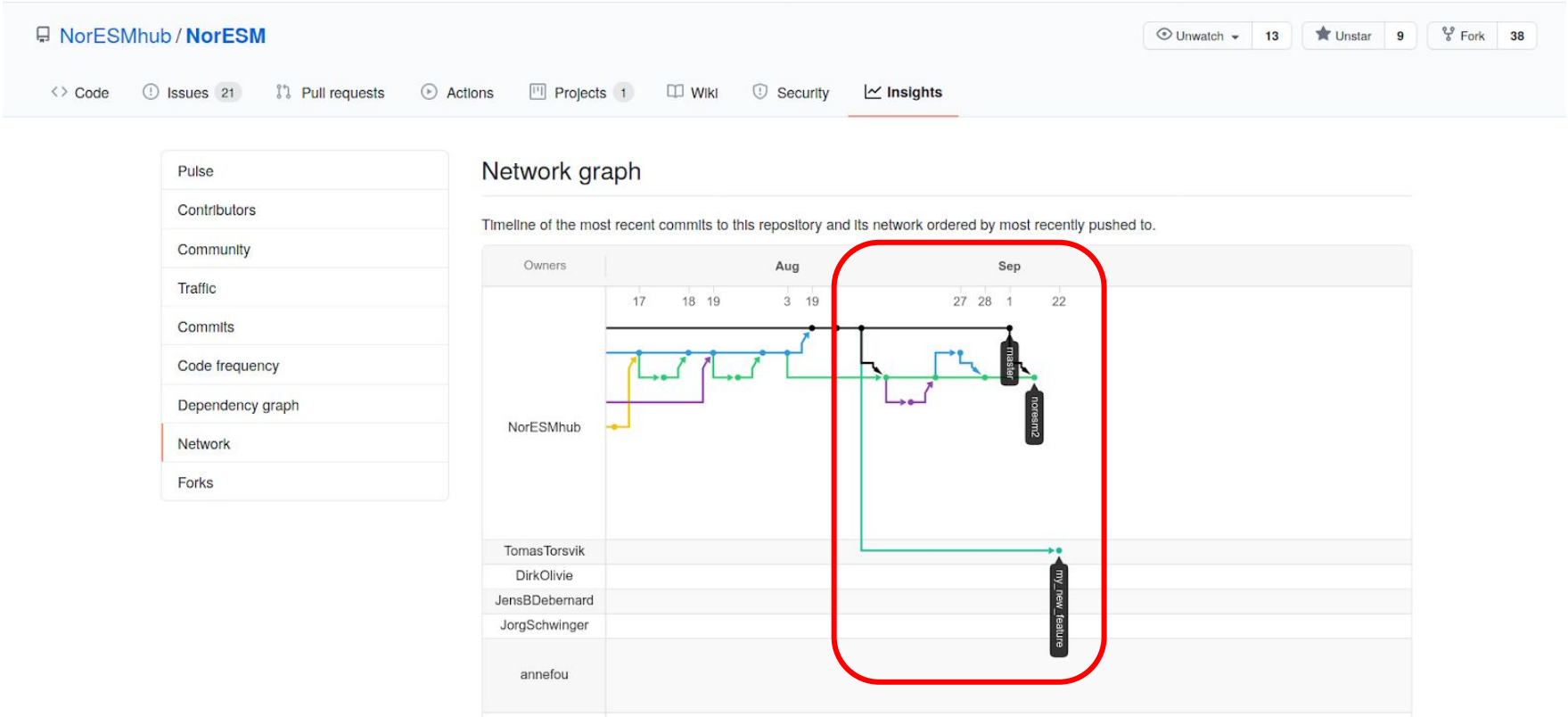


- Create a new feature branch before making any changes
  - Allows updating of root branch (e.g. “master”) without interfering with your own work
- Apply your changes in the feature branch, either directly in the gitHub fork or your local clone of the fork

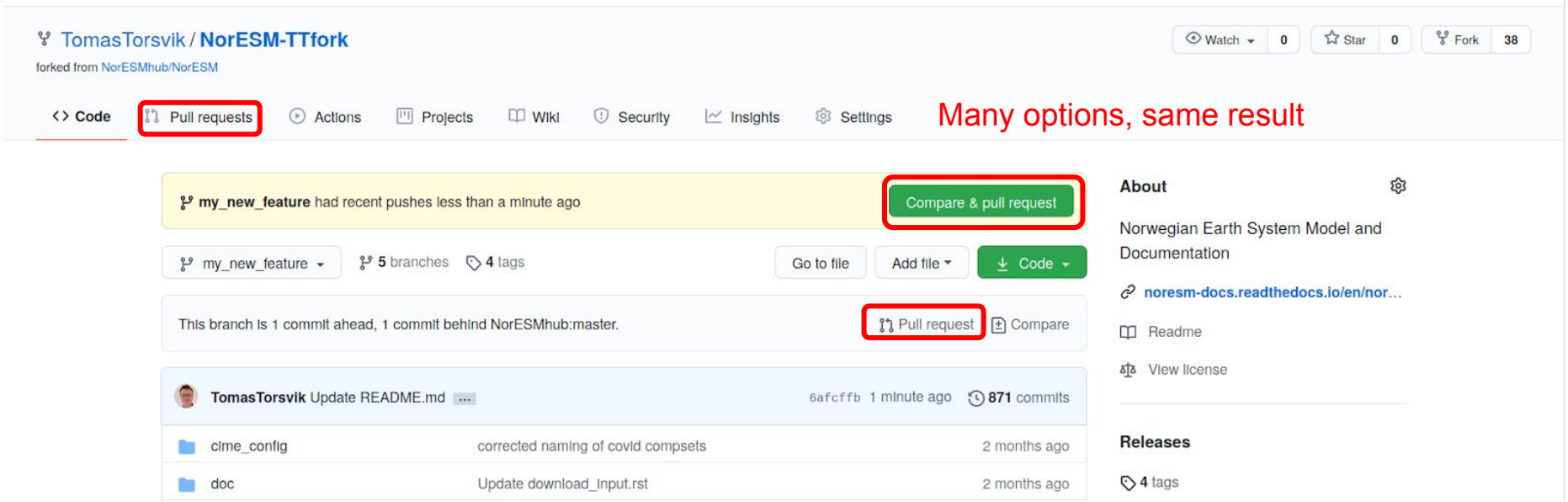
Review changes : from fork



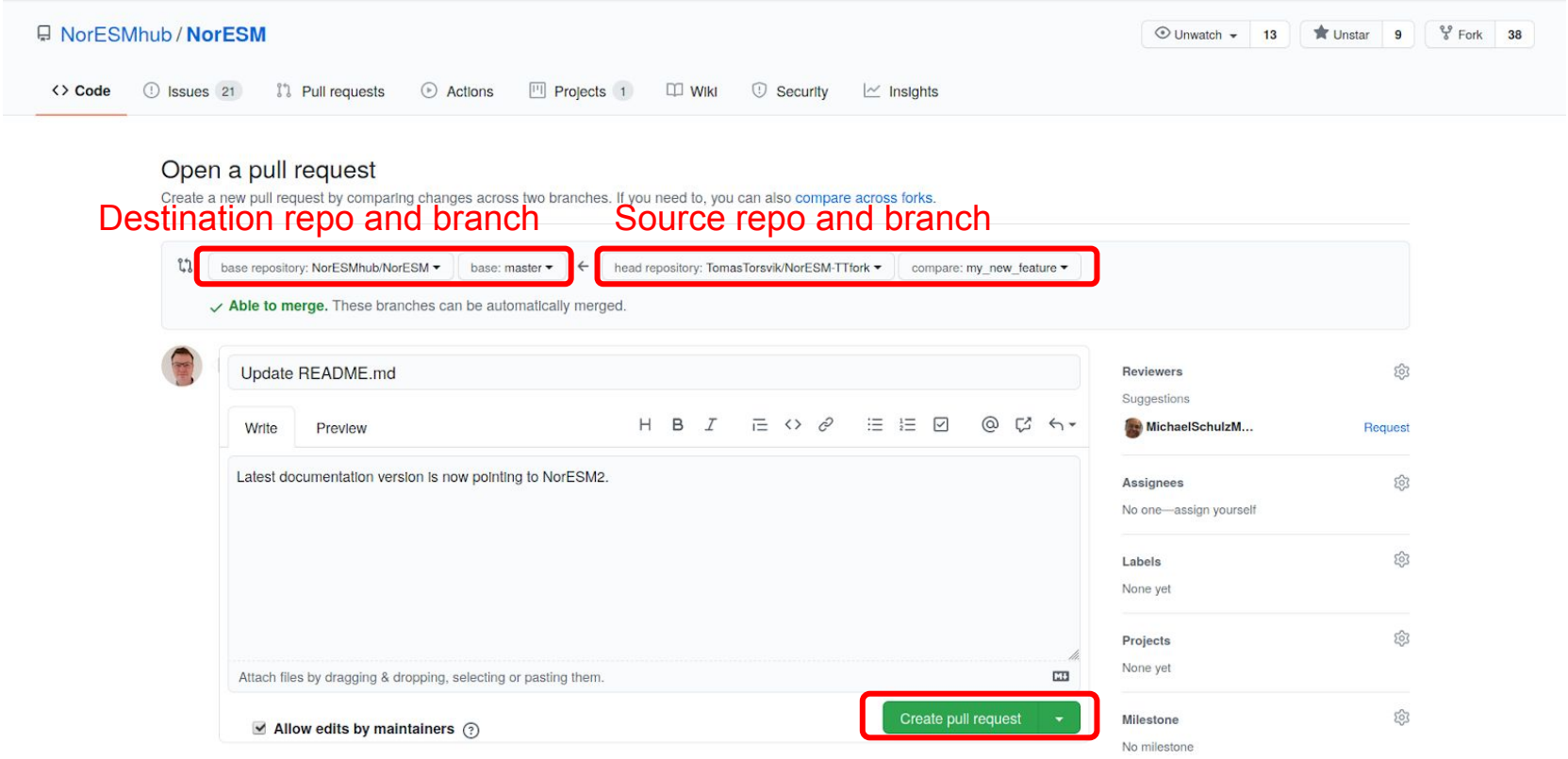
Review changes : from “Network”



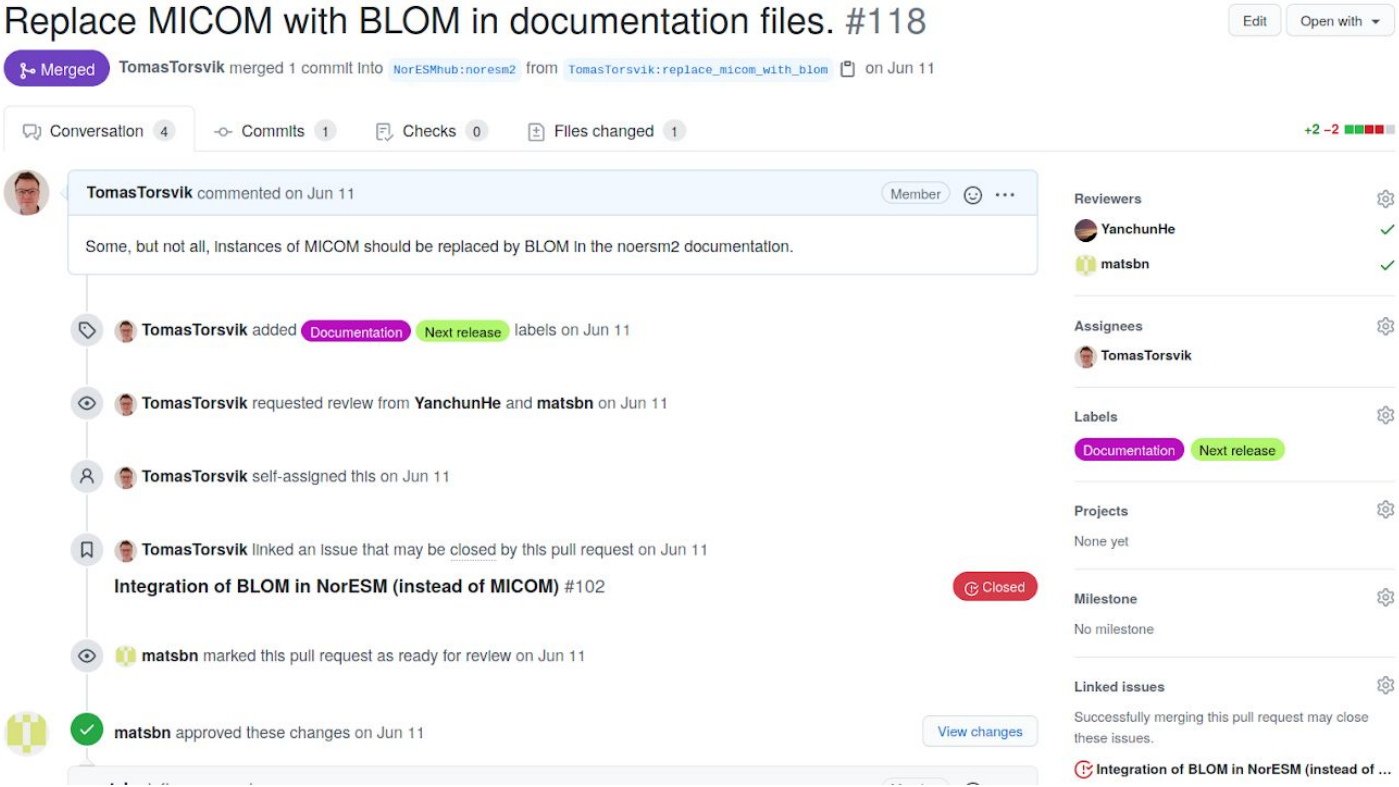
Create a pull request



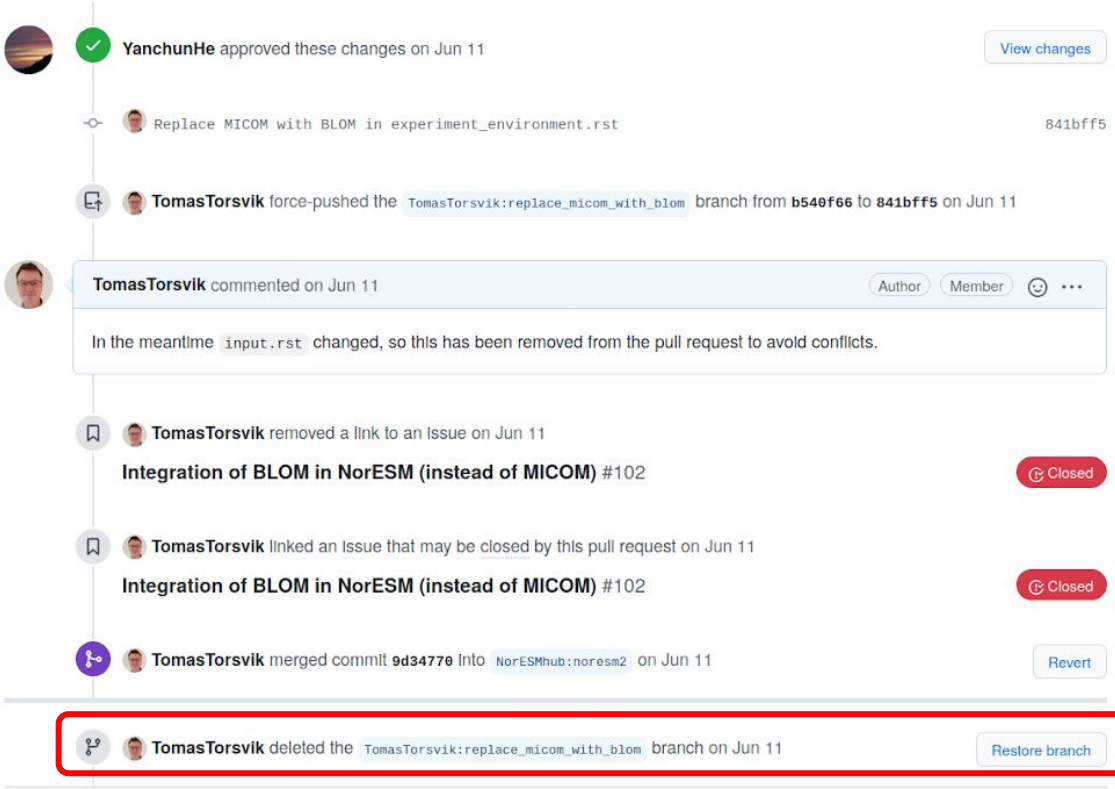
Create a pull request



Pull request : completed example

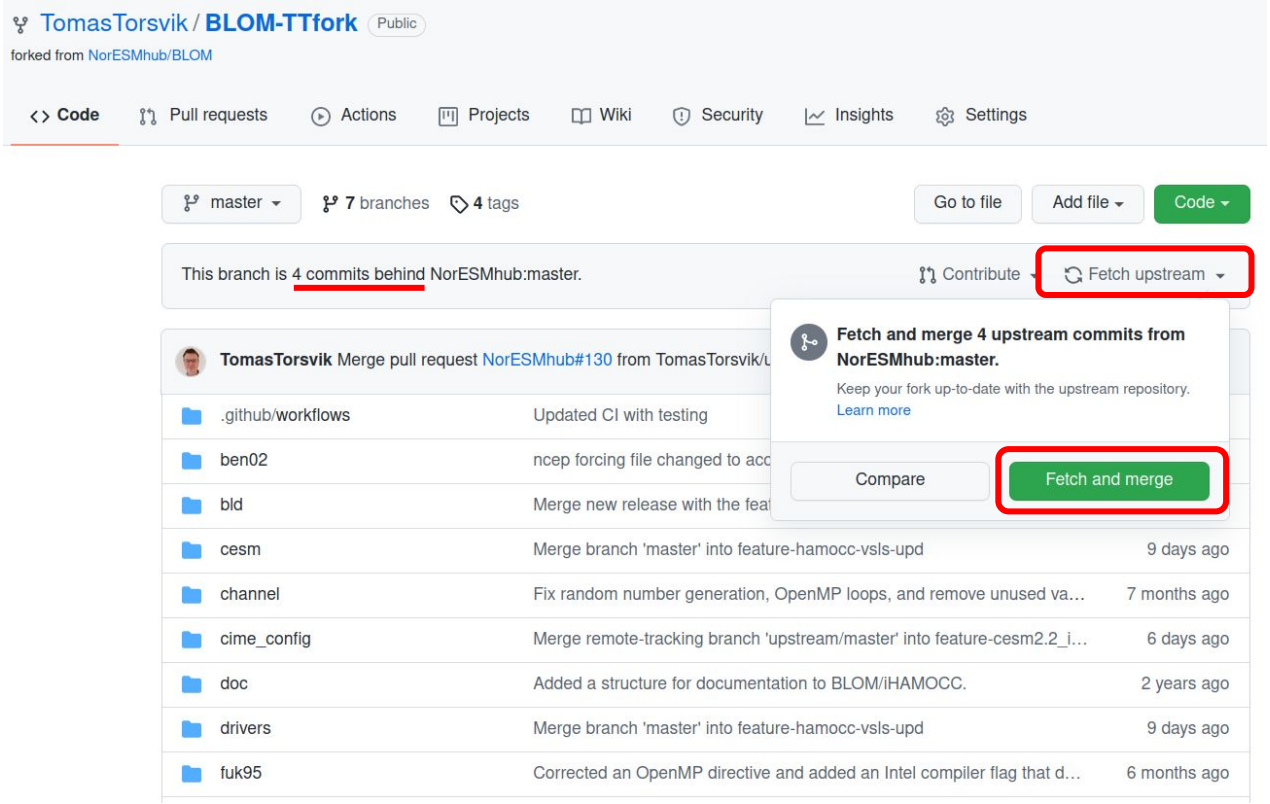


Pull request : completed example



Delete “throw away” branch
In personal fork

Keep fork in sync with main repository



The “Fetch and merge” option will bring the “master” branch in sync with the main repository ...

-- without conflicts or additional merge commits --

... if you have not made any changes in the “master” branch for your own fork.

Include your own code changes (Tomas)

- Run NorESM with code modifications

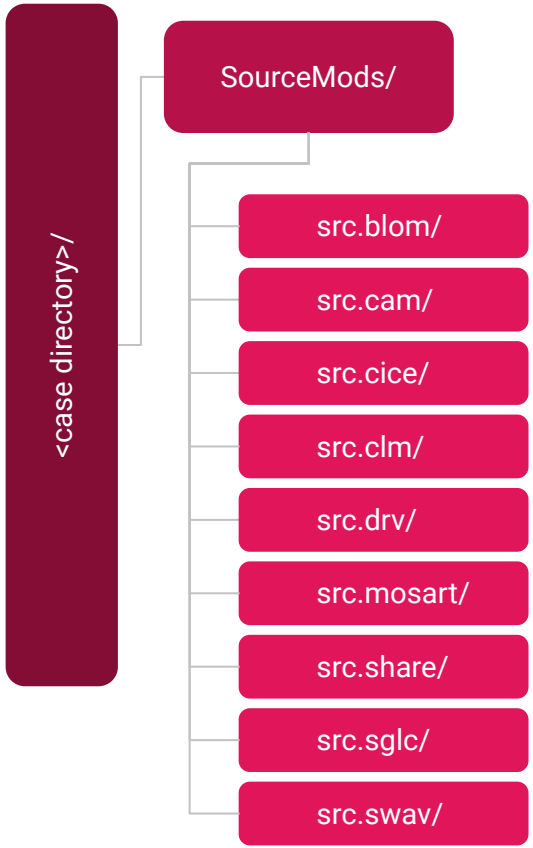
How to include code modifications

1. Include changes directly in a case SourceMods directory
    - Suitable for testing small code changes and debugging.

2. Copy existing SourceMods into your case directory
    - Used for some standard NorESM2 model configs, preferably included when the case is created.

3. Include link to modified source files that exist in a git/github repository
    - This is the most consistent option when working with model (component) development

Include your own SourceMods



Each model component (also data and stub components) have a dedicated sub-directory under SourceMods/ where a user can include their own source modification.

Steps to use SourceMods:

1. Copy a file (subroutine, module) from a NorESM component source directory into the corresponding src.<component> directory.
2. Make changes in the SourceMods file
3. Build and run the modified model using standard scripts (./case.build , ./case.submit)

Include pre-defined sourceMods for piControl run

NorESM2-MM piControl :
./create_newcase --case <casename> --compset N1850 --res f09_tn14 --machine betzy --project <nn????k> --user-mods-dir cmip6_noresm_DECK

- In case directory, change RUN_TYPE to branch, RUN_REFCASE to the CMIP6 piControl casename, and RUN_REFDATE to the start of the piControl experiment

./xmlchange RUN_TYPE=branch
./xmlchange RUN_REFCASE=N1850_f09_tn14_20190913
./xmlchange RUN_REFDATE=1200-01-01

- Copy restart and rpointer files to run directory, and unzip files:

cp /trd-project3/NS9560K/noresm/cases/N1850_f09_tn14_20190913/rest/1200-01-01-00000/* /cluster/projects/$PROJECT/$USER/noresm/<casename>/run/
gunzip /cluster/projects/$PROJECT/$USER/noresm/<casename>/run/*.gz

Documentation: Reproduce CMIP6 piControl, historical and SSP5-8.5 experiments
https://noresm-docs.readthedocs.io/en/latest/configurations/cmip6_compsets.html?highlight=user-mods-dir#reproduce-cmip6-picontrol-historical-and-ssp5-8-5-experiments

Configure External.cfg file

```
[cam]
tag = cam_cesm2_1_rel_05-Nor_v1.0.4
protocol = git
repo_url = https://github.com/NorESMhub/CAM
local_path = components/cam
required = True

---

[clm]
tag = release-clm5.0.14-Nor_v1.0.3
protocol = git
repo_url = https://github.com/NorESMhub/ctsm
local_path = components/clm
externals = Externals_CLM.cfg
required = True
```

Format

[component name]

One of the following:
- tag = checkout tag
- hash = checkout commit hash (only git)
- branch = a branch from the specified repository

protocol = [git, svn]
repo_url = location of external source repository
[github.url] or [local/path]
local_path = where to place local clone of source
externals = (optional) sub-externals required by the specific component
required = is the component required?
[true, false]

Configure External.cfg file

Source from github:

```
[blom]
branch = bugfix_write_tracer_restart
protocol = git
repo_url = https://github.com/TomasTorsvik/BLOM-TTfork
local_path = components/blom
required = True
```

Source from local repository:

```
[blom]
branch = master
protocol = git
repo_url = ${HOME}/nn2980k/tomast/BLOM/BLOM-TTfork
local_path = components/blom
required = True
```

Externals.cfg can be changed to download source files from a fork repository on github or a locally cloned repository.

`checkout_externals` will complain if your repository is not in a clean state (includes unmerged changes), but you can still run `create_newcase` with the unmerged files.

`checkout_externals` will NOT automatically update branches that have been changed remotely on github. Update these manually

```
git fetch
git checkout origin/<branch>
```

Hands-on session 4

Try to fetch source code from a different location than defined by Externals.cfg

1. Create a copy of a NorESM component to store your own code modifications. (do one of the following)
   a. Create a fork of a NorESM component in your own github account
   b. Clone a NorESM component from github
2. Make a copy of Externals.cfg for your modifications, e.g.
   `cp Externals.cfg My_Externals.cfg`
3. Edit My_Externals.cfg to point to your own source (repo_url) and optionally a different tag/branch/hash
4. Run `./manage_externals/checkout_externals -e My_Externals.cfg` and confirm that your modified component has been included in NorESM