CESM1/CCSM4 Tutorial: Modifications

Part 1: Namelist Modifications

Part 2: Code Modifications

Part 3: Exercises and Solutions

Cecile Hannay, CAM Science Liaison
Atmospheric Modeling and Predictability Section
Climate and Global Dynamics Division

NSF NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH U.S. DEPARTMENT OF ENERGY Office of Science

Part 1: Namelist Modifications

Part 2: Code Modifications

Part 3: Exercise Solutions

Namelists mods: What do already know ?

Reminder from day 2:

There are 2 ways to modify the CESM namelists:

Editing : ***env_conf.xml***

CAM_NAMELIST_OPTS → CAM namelist options
CLM_NAMELIST_OPTS → CLM namelist options
CICE_NAMELIST_OPTS → CICE namelist options

Day 2

Editing: ***Buildconf/$component.buildnml.csh***

Buildconf/cam.buildnml.csh → $RUNDIR/atm_in
Buildconf/clm.buildnml.csh → $RUNDIR/lnd_in
Buildconf/pop2.buildnml.csh → $RUNDIR/pop2_in
Buildconf/cice.buildnml.csh → $RUNDIR/ice_in
Buildconf/cpl.buildnml.csh → $RUNDIR/drv_in

Day 2 and today

Part I: Namelist modifications

• The configure command generates *buildnml* files for each component:
Buildconf/$component.buildnml.csh

• You can customize your namelists by editing these files ** and running the build script:
$CASE.$MACH.build
This will create customized namelist files in your run directory:
atm_in, drv_in, ice_in, lnd_in, pop2_in

• If you issue the commands:
configure –cleannamelist or *configure -cleanall*
all your changes are gone !!!

*** In this tutorial, most examples will be coming from the atmospheric model*

Overview of namelist modifications

In this section, we will cover:

- how to change the output frequency
- how to output extra variables
- how to output extra history files
- how to output a set of columns
- how to control the number of time samples written to a history file

Customizing CAM history files: nhtfrq, mfilt

The default history file from CAM is a monthly average.

We can change the output frequency with the namelist variable ***nhtfrq***

If nhtfrq=0, the file will be a monthly average
If nhtfrq>0, frequency is input as number of timesteps.

If nhtfrq<0, frequency is input as number of hours.

For instance to change the history file from monthly average to daily average, we set the namelist variable:
nhtfrq = -24

To control the number of timestep in the history file, we can use the variable ***mfilt***
For instance, to specify that we want one time sample on each history file, we set the namelist variable:

mfilt = 1

Customizing CAM history files: fincl

You can output up to 6 history files: **“h0”, “h1”, …, “h5”.**

The file “h0” contains the default variables (in the code: “call add_default”). This includes the variables necessary for the AMWG package.

For the files “h1” to “h5”, the user has to specify the variables to output.

We can use the namelist variables ***fincl1, fincl2,.., fincl6*** to control the list of fields in the history files: **h0 h1 h5**

The added fields must be in Master Field List (= fields that can be written to the history files).

Using a ":" following a field gives the **averaging flag** for the output field. Valid flags are: I for instantaneous, A for average, M for minimum, and X for maximum.

Example of customizing history files

For instance, on the top of the monthly history file “h0”, if we want to output a file “h1” with instantaneous values of T, Q, U, V and OMEGA every 3 hour, we can use:

fincl2 = 'T:I','Q:I','U:I','V:I',‘OMEGA:I’
nhtfrq = 0, -3

Notice that it is equivalent to:

fincl2 = 'T:I','Q:I','U:I','V:I’, ‘OMEGA:I’
nhtfrq(1) = 0
nhtfrq(2) = -3

NB: If you plan to run the AMWG diagnostic package, it is recommended to leave the “h0” file untouched and to add extra history files.

Output a set of columns: fincl1lonlat

fincl1lonlat: allows to output a column or set of contiguous columns for fincl1 variables

Output a single column:

fincl1 = 'T','Q','U','V'

fincl1lonlat = '180e_60n'

Output an area:

fincl1 = 'T','Q','U','V'

fincl1lonlat = '170e:240e_30n:60n'

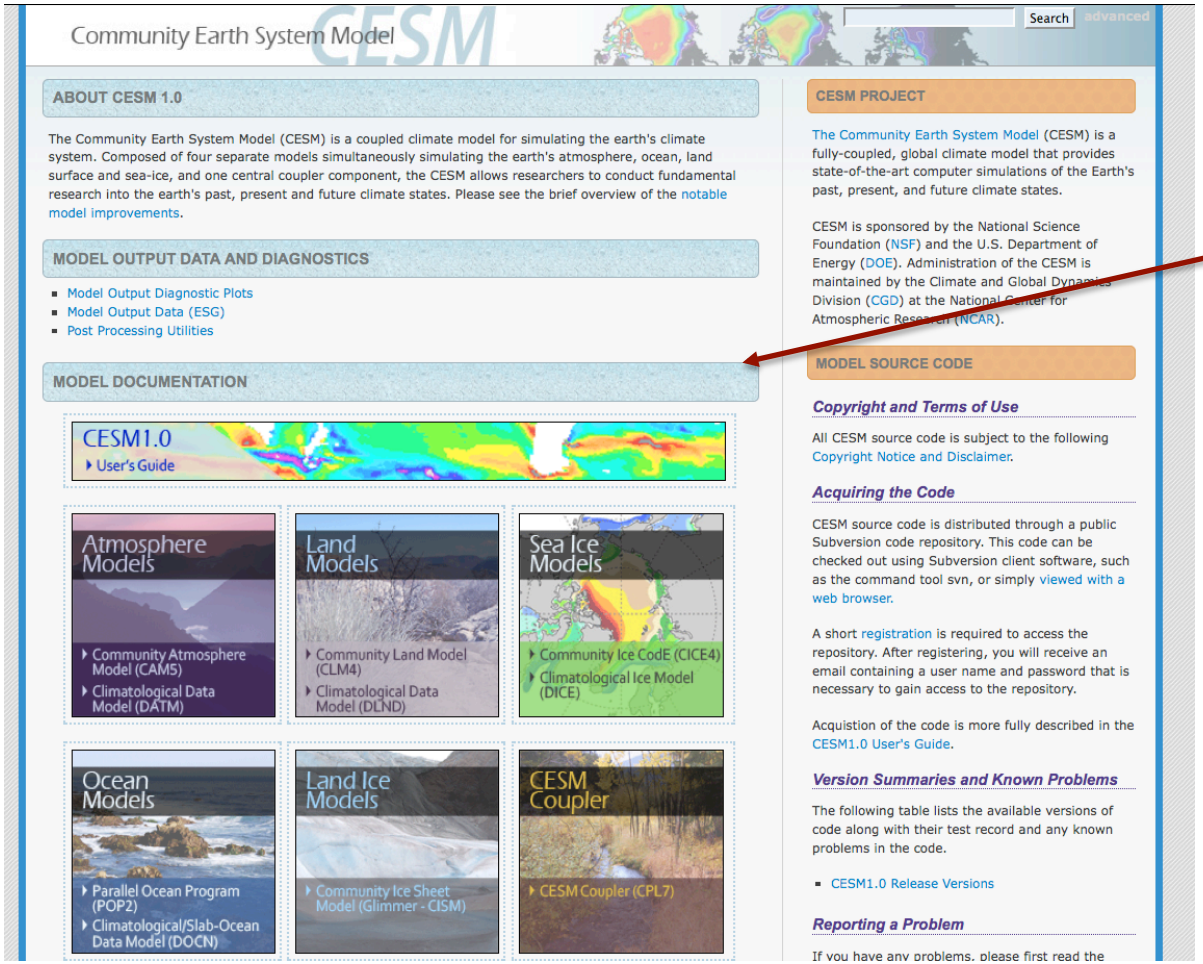
Output combination of columns and/or areas:

fincl1 = 'T','Q','U','V'

fincl1lonlat = '180e_60n', '170e:240e_30n:60n'

Where to find the documentation ?

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



Model documentation for each component of CESM

CAM namelist documentation ?

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

CESM | Administration | Working Groups | Models | Events | News | Publications

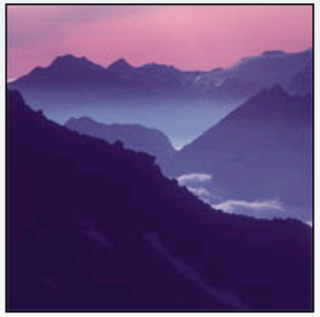
Community Earth System Model

Search advanced

CESM1.0: CAM DOCUMENTATION

Introduction

Version 5.0 of the Community Atmosphere Model (CAM) is the latest in a series of global atmosphere models developed primarily at the National Center for Atmospheric Research (NCAR). CAM5.0 includes significant enhancement to the represntation of atmopsheric processes resulting a a number of notable improvements. CAM4.0 is also available in the CESM1.0 release. Development of the model was led by the Atmosphere Model Working Group (AMWG).



Documentation

CAM5

- Users Guide [html (single page)] [html (multipage)] [pdf]
- Search Namelist Variables
- Scientific description (*coming soon*) [ps] [pdf]

CAM4

- Users Guide [html (single page)] [html (multipage)] [pdf]
- Search Namelist Variables
- Scientific description [ps] [pdf]

Search Namelist Variables

CAM namelist documentation ?

Search or Browse CAM Namelist Variables

This page contains the complete list of namelist variables available in CAM-4.0. They are grouped by categories designed to aid browsing. Clicking on the name of a variable will display descriptive information. If search terms are entered in the text box below, the list will be condensed to contain only matched variables.

Search Variable Names | Show All Variable Names

AND OR (separate search terms with spaces)

Also search help text

Search Variables Names or Show All Variables Names

Control - Driver

Namelist Variable	Type	Group
aqua_planet	logical	seq_infodata_inparm
atm_adiabatic	logical	seq_infodata_inparm
atm_ideal_phys	logical	seq_infodata_inparm
atm_logfile	char*256	camexp
atm_logfile_diro	char*256	camexp
atm_ntasks	integer	ccsm_pes
atm_nthreads	integer	ccsm_pes
atm_pestride	integer	ccsm_pes
atm_rootpe	integer	ccsm_pes
bfbflag	logical	seq_infodata_inparm
brnch_retain_casename	logical	seq_infodata_inparm
budget_ann	integer	seq_infodata_inparm
budget_daily	integer	seq_infodata_inparm
budget_inst	integer	seq_infodata_inparm
budget_ltann	integer	seq_infodata_inparm
budget_ltend	integer	seq_infodata_inparm
budget_month	integer	seq_infodata_inparm
case_desc	char*256	seq_infodata_inparm

CLM namelist documentation ?

http://www.cesm.ucar.edu/models/cesm1.0/clm/models/lnd/clm/doc/UsersGuide/x1827.html

Customizing the CLM namelist

Once a case is **configured**, we can then customize the case further, by editing the run-time namelist for CLM. First let's list the definition of eac
list the default values for them. Next for some of the most used or tricky namelist items we'll give examples of their use, and give you example

Definition of Namelist items and their default values

Here we point to you where you can find the definition of each namelist item and separately the default values for them. The default values ma
simulation-year and other attributes. Both of these files are viewable in your web browser. Below we provide the link for them, and then explai

1. Definition of each Namelist Item
2. Default values of each CLM Namelist Item

Definition and values of namelist Variables

Definition of CLM namelist variables

Note, these all would go into the clm.buildnml.csh file:

Included in the table are the following pieces of information:

- Variable name.
- Variable type (char, integer, real, or logical). The type char has the length appended following an asterisk, e.g., char*256. Variables that a
inside parentheses. For example char*1(6) denotes a array of six char*1 values.

CICE namelist documentation ?

http://www.cesm.ucar.edu/models/cesm1.0/cice/doc/index.html

Contents

1 Introduction 2
1.1 What's new in CICE4? 2

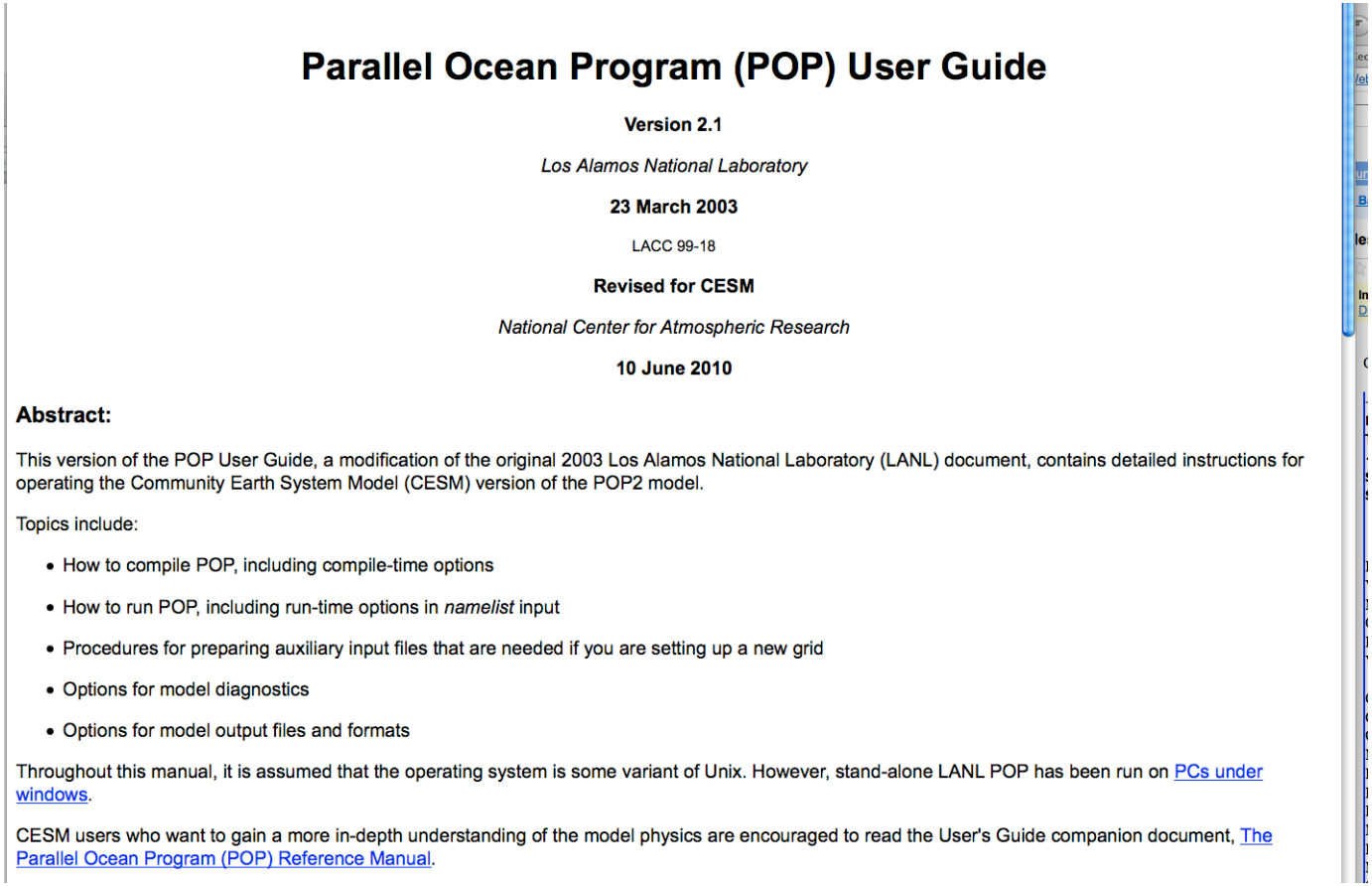
2 The CICE Scripts 3
2.1 Coupled Model Scripts 4
2.2 The Build Environment 4
2.2.1 CICE Preprocessor Flags 5

3 Namelist Variables 5
3.1 Changing the timestep 6
3.2 Writing Output 7
3.3 Model Physics 7
3.4 Tracer Namelist 9
3.5 Prescribed Ice Namelist 9
3.6 Grid Namelist 9
3.7 Domain Namelist 10
3.8 PIO Namelist 10
3.9 Example Namelists 10
3.9.1 Example 1: CESM Fully Coupled 11
3.9.2 Example 2: History File Namelist 12

Section on Namelist Variables

POP2 namelist documentation ?

http://www.cesm.ucar.edu/models/cesm1.0/pop2/doc/users/POPusers_main.html



Parallel Ocean Program (POP) User Guide

Version 2.1

Los Alamos National Laboratory

23 March 2003

LACC 99-18

Revised for CESM

National Center for Atmospheric Research

10 June 2010

Abstract:

This version of the POP User Guide, a modification of the original 2003 Los Alamos National Laboratory (LANL) document, contains detailed instructions for operating the Community Earth System Model (CESM) version of the POP2 model.

Topics include:

- How to compile POP, including compile-time options
- How to run POP, including run-time options in *namelist* input
- Procedures for preparing auxiliary input files that are needed if you are setting up a new grid
- Options for model diagnostics
- Options for model output files and formats

Throughout this manual, it is assumed that the operating system is some variant of Unix. However, stand-alone LANL POP has been run on PCs under windows.

CESM users who want to gain a more in-depth understanding of the model physics are encouraged to read the User's Guide companion document, The Parallel Ocean Program (POP) Reference Manual.

Namelist modifications: Exercises

Exercise 1

Using the compset *B_1850_CN*, make one-month run:
- Output daily averages for atm.
- Write one history file for every day of the month.

(Hint: Use namelist variables: *nhtfrq, mfilt*)

Exercise 2

Using the compset: B_1850_CN, make one-month run:
On the top of the monthly history file “h0”, output:

- “h1” file with instantaneous values of T, Q, U and V every 3 hour.
- “h2” file with time-average values of T, Q, U and V every 24 hour.

Write one h1 file for every day of the month and write a single h2.
(Hint: Use namelist variables: *nhtfrq, mfilt, fincl*)

Namelist modifications: Exercises

Exercise 3

Using the compset: F_2000 (CAM standalone, prescribed ocn/ice, present day run)
Make three-month run: output a h1 history with your favorite set of variables over the storm track area

(Hint: Use namelist variables: *fincl2lonlat, fincl2*)

Solutions

Solutions to the exercises are at the **end of the tutorial**.

We will go over exercise 1 as an example.

You can try to do the exercises on your own using the namelist documentation (*recommended*).

Exercise 1

Using the compset *B_1850_CN*, make one-month run:
- Output daily averages for atm.
- Write one history file for every day of the month.

• Setting environment for this exercise:

setenv CASENAME ex1
setenv CESMROOT /fis/cgd/cseg/csm/collections/cesm1_0_tutorial
setenv CASEDIR /blhome/{$LOGNAME}/cases/$CASENAME
setenv ARCHIVEDIR /ptmp/{$LOGNAME}/archive/$CASENAME
setenv RUNDIR /ptmp/{$LOGNAME}/$CASENAME

The name of your case
This is where the CESM code lives
Your case directory
The short-term archive for this case
Your run directory

NB: You are in charge. Feel free to set *CASENAME* and *CASEDIR* to whatever you want.

Exercise 1

• Create a new case:

```
cd $CESMROOT/scripts
./create_newcase -case $CASEDIR \
                 -mach bluefire  \
                 -res T31_gx3v7  \
                 -compset B_1850_CN
```

• Configure the model:

```
cd $CASEDIR
./configure -case
```


Exercise 1

- Examine the contains of the directory: ***$CASEDIR/Buildconf***

```
cam.buildexe.csh        ciceconf                mct.buildlib
cam.buildnml.csh        clm.buildexe.csh        pio.buildlib
cam.input_data_list     clm.buildnml.csh        pop2.buildexe.csh
camconf                 clm.input_data_list     pop2.buildnml.csh
ccsm.buildexe.csh       clmconf                 pop2.input_data_list
cice.buildexe.csh       cpl.buildexe.csh        pop2doc
cice.buildnml.csh       cpl.buildnml.csh        sglc.buildexe.csh
cice.input_data_list    csm_share.buildlib      sglc.buildnml.csh
```

- Examine the file: ***$CASEDIR/Buildconf/cam.buildnml.csh***

```
&cam_inparm
 fincl1   = 'U'
 fincl2   = 'TREFHTMN','TREFHTMX','TREFHT','PRECC','PRECL','PSL'
 nhtfrq   = 0,-24,-24,-6,-3
 mfilt    = 1,365,30,120,240
```

This variable is added to file “h0”

Create a file “h1” with these variables

Monthly output for “h0” and daily output for “h1”

1 time sample for “h0” and 365 time samples for “h1”

Not used

Exercise 1

- Edit: ***$CASEDIR/Buildconf/cam.buildnml.csh***

```
&cam_inparm
 nhtfrq    = -24
 mfilt     = 1
 fincl2    = 'TREFHTMN','TREFHTMX','TREFHT','PRECC','PRECL','PSL'
```

Daily averages for h0

one time sample per day for h0

Delete this line

- Edit: ***$CASEDIR/env_run.xml***

cd $CASEDIR

xmlchange -file env_run.xml -id STOP_N -val '1'

xmlchange -file env_run.xml -id STOP_OPTION -val 'nmonths'

Exercise 1

- Build

cd $CASEDIR

./{$CASENAME}.bluefire.build

- Edit the run script*:* ***$CASEDIR/{$CASENAME}.bluefire.run***

#BSUB –U 37591051#4 (if available; not for auditors)

#BSUB –W 1:50

- Submit the job

cd $CASEDIR

bsub < {$CASENAME}.bluefire.run

- Check the job is running

bjobs

Exercise 1

• When job is completed, look at the output files:

cd $ARCHIVEDIR/atm/hist

ls

```
ex1.cam2.h0.0001-01-01-00000.nc  ex1.cam2.h0.0001-01-12-00000.nc  ex1.cam2.h0.0001-01-23-00000.nc
ex1.cam2.h0.0001-01-02-00000.nc  ex1.cam2.h0.0001-01-13-00000.nc  ex1.cam2.h0.0001-01-24-00000.nc
ex1.cam2.h0.0001-01-03-00000.nc  ex1.cam2.h0.0001-01-14-00000.nc  ex1.cam2.h0.0001-01-25-00000.nc
ex1.cam2.h0.0001-01-04-00000.nc  ex1.cam2.h0.0001-01-15-00000.nc  ex1.cam2.h0.0001-01-26-00000.nc
ex1.cam2.h0.0001-01-05-00000.nc  ex1.cam2.h0.0001-01-16-00000.nc  ex1.cam2.h0.0001-01-27-00000.nc
ex1.cam2.h0.0001-01-06-00000.nc  ex1.cam2.h0.0001-01-17-00000.nc  ex1.cam2.h0.0001-01-28-00000.nc
ex1.cam2.h0.0001-01-07-00000.nc  ex1.cam2.h0.0001-01-18-00000.nc  ex1.cam2.h0.0001-01-29-00000.nc
ex1.cam2.h0.0001-01-08-00000.nc  ex1.cam2.h0.0001-01-19-00000.nc  ex1.cam2.h0.0001-01-30-00000.nc
ex1.cam2.h0.0001-01-09-00000.nc  ex1.cam2.h0.0001-01-20-00000.nc  ex1.cam2.h0.0001-01-31-00000.nc
ex1.cam2.h0.0001-01-10-00000.nc  ex1.cam2.h0.0001-01-21-00000.nc  ex1.cam2.h0.0001-02-01-00000.nc
ex1.cam2.h0.0001-01-11-00000.nc  ex1.cam2.h0.0001-01-22-00000.nc
```

• Examine the output variables and the number of time samples in a history file:

cd $ARCHIVEDIR/atm/hist

ncdump -h {$CASENAME}.cam2.h0.0001-01-01-00000.nc

ncdump -v time {$CASENAME}.cam2.h0.0001-01-01-00000.nc

Part 1: Namelist Modifications

Part 2: Code Modifications

Part 3: Exercise Solutions

NSF
NCAR NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
U.S. DEPARTMENT OF ENERGY | Office of Science

Part II: Code modifications

This section gives an overview of simple code modifications

– Modifying a parameter in the code

– Adding an output field from a variable

Principles for modifying the code

Never modify the CESM root itself. Your modifications to the code should go into: ***SourceMods***

SourceMods contains subdirectories for each component:

- *src.cam* → because we are looking at CAM, this is where we put our mods
- *src.cice*
- *src.clm*
- *src.docn*
- *src.drv*
- *src.sglc*
- *src.share*

Modifying a subroutine

- **Steps to modify the code**:
  - Find the subroutine you want to modify
  - Copy this subroutine in SourceMods
  - Make your mods
  - Compile and run the model

- **Tools to find the subroutine you want**:

  grepccm: walk through a list of directories specified in the Filepath file grepping for a string within the files of those directories

  findccm: walk through a list of directories specified in the Filepath file looking for a file

  where: **Filepath** is a file containing the path to all your subroutines

/fis/cgd/home/hannay/bin/grepccm

Tools: grepccm/findccm

- **Filepath**

Filepath (for atm) is located in your rundir: *$RUNDIR/atm/obj*

```
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/chemistry/bulk_aero
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/chemistry/utils
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/physics/cam
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/dynamics/eul
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/cpl_mct
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/control
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/utils
/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/advection/slt
```

- **Location for grepccm and findccm**

/fis/cgd/home/hannay/bin/grepccm

/fis/cgd/home/hannay/bin/findccm

Example: How to use grepccm/findccm ?

- Say we want to find the string "Moist entrainment" in CAM.

- Go into the directory where Filepath is:

cd $RUNDIR/atm/obj

grepccm "Moist entrainment"

The string "Moist entrainment" is in the subroutine eddy_diff.F90

---- searching /ptmp/hannay/cases/ex4/SourceMods/src.cam

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/chemistry/bulk_aero

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/chemistry/utils

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/physics/cam

eddy_diff.F90: real(r8), parameter :: a2l = 30._r8 ! Moist entrainment enhancement param (recommended range : 10~30)

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/dynamics/eul

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/cpl_mct

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/control

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/utils

---- searching /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/advection/slt

findccm eddy_diff.F90

The subroutine eddy_diff.F90 is in this directory

found in /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/physics/cam

Output an extra variable

• One common thing is to output a variable that is not already output from the model

• For instance, in CAM:
- there are fields for total and ice in-cloud water paths:
  ICLDIWP (ice) and ICLDTWP (liquid + ice)
- but there is no field for liquid in-cloud water path
- It is easy to make one: ICLDLWP

This can be done by a succession of calls:

call addfld ('ICLDLWP', …) → Add a field to master field list

call add_default ('ICLDLWP',…) → Add this field to "h0" by default (optional)

call outfld('ICLDLWP', …) → Collect values for this field and write to history file

Syntax: addfld

addfld = Add a field to master field list

Field name

Units

Number of vertical levels:
surface :1
Multi-level: pver or pverp

Averaging flag:
A = average
I = instantaneous

Field full name

```
subroutine addfld (fname, units, numlev, avgflag, long_name, &
    decomp_type, [flag_xyfill], [flag_isccplev], [sampling_seq])
```

Decomposition type (phys_decomp or dyn_decomp)

Optional (not typical) non-applicable xy points flagged with fillvalue

Optional (not typical) ISCPP simulator level

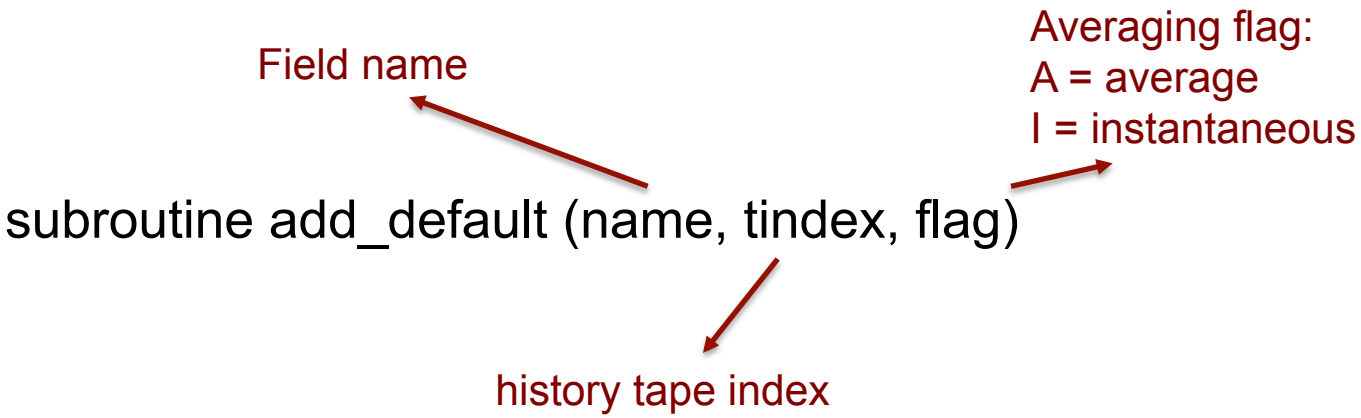
Optional (typical): indicates how often field is output. The default is every timestep. Only valid value is 'rad_lwsw' (field sampled only during LW/SW radiation calculation)

Example:

```
call addfld ('ICLDIWP', 'gram/m2', pver, 'A','In-cloud ice water path' ,
     phys_decomp, sampling_seq='rad_lwsw')
```


Syntax: add_default

add_default = Add a field to the list of default fields on history file

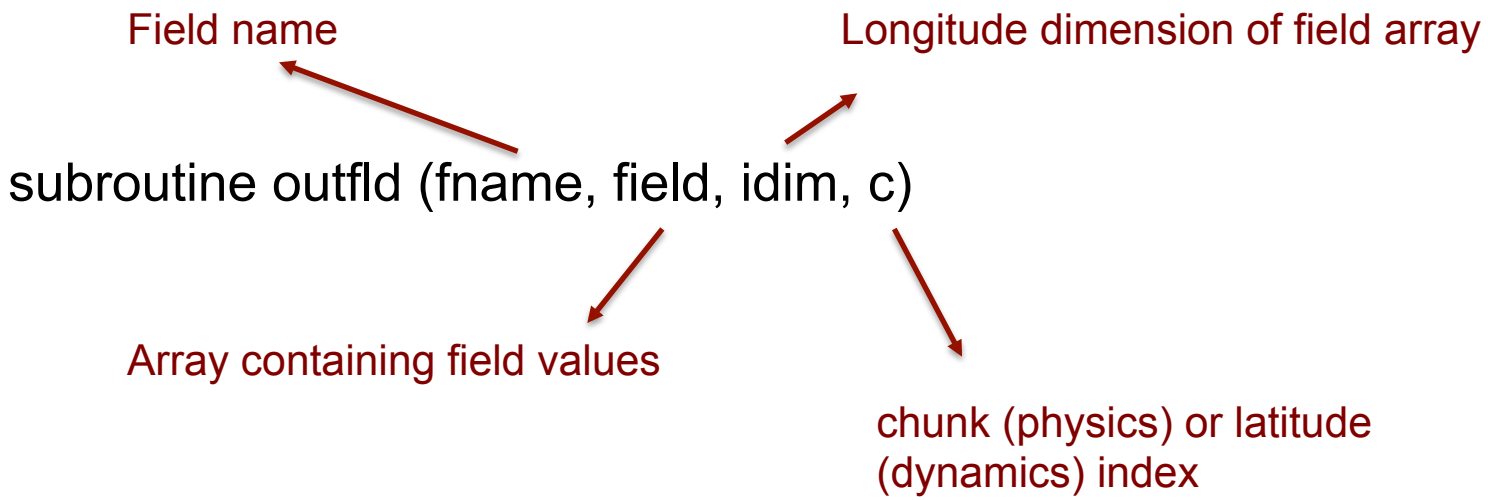


Example:

call add_default ('ICLDIWP', 1, ' ')

Syntax: outfld

add_default = Accumulate (or take min, max, etc. as appropriate) input field into its history buffer for appropriate tapes



Example:

call outfld('ICLDIWP' ,cicewp , pcols,lchnk)

Code modifications: Exercises

Exercise 4

Using the compset: B_1850_CN, add a variable for the liquid in-cloud water path: ICLDLWP and make a 1-month run.
(Hint: Use ICLDIWP as a template for your changes.
Find the subroutine containing ICLDIWP using grepccm/findccm)

Exercise 5

Using the compset: B_1850_CN and make a 1-month run.
Change the value of Dcs (autoconversion size threshold for cloud ice to snow) to Dcs = 300.e-6_r8

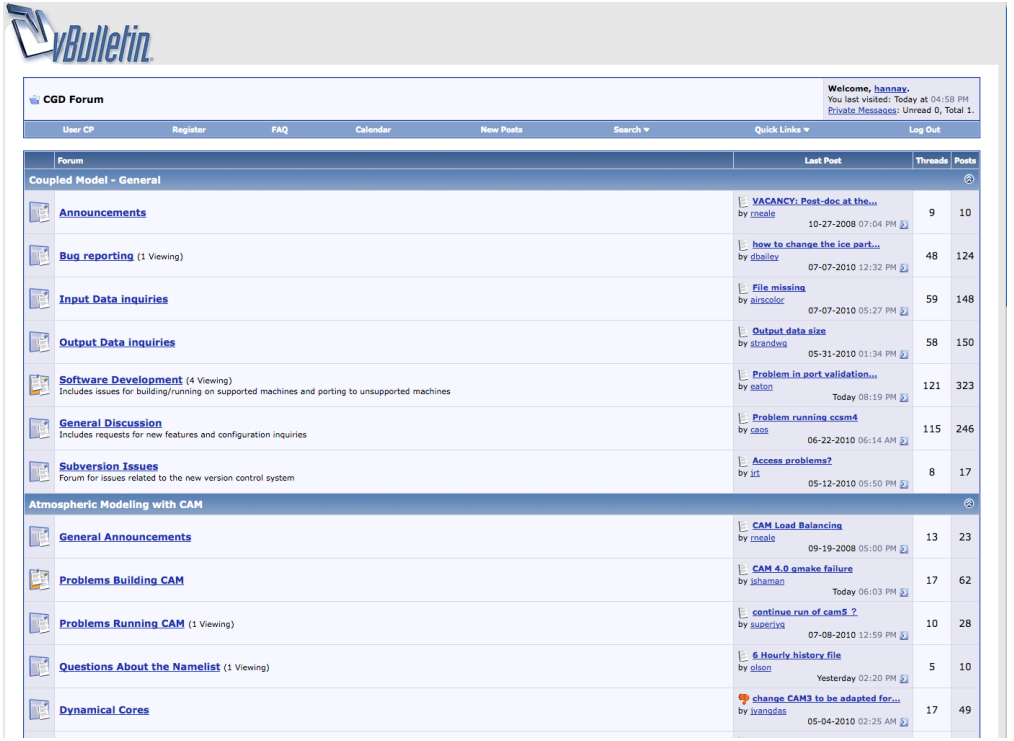
If you do more elaborate mods

- Know what your are doing
- Understand the structure of the code

Where to find help ?

Documentation:
http://www.cesm.ucar.edu/models/cesm1.0/index.html

CESM bulletin board:
http://bb.cgd.ucar.edu/



Part 1: Namelist Modifications

Part 2: Code Modifications

Part 3: Exercise Solutions

Suggestions

Try to do the exercises on your own using the hints and the online documentation. Look at the solutions if your are stucked.

Document everything you do in the README.case file

You are in charge. In the exercises, feel free to set *CASENAME* and *CASEDIR* to the values you want.

If you are running out of time, try to do one exercise with namelists modifications (1,2 or 3) and one exercise with the source modifications (4 or 5)

Exercise 1

• Using the compset *B_1850_CN* (all active components, pre-industrial, with CN in CLM), make one-month run:
- Output daily averages for atm.
- Write one history file for every day of the month.

(Hint: Use the namelist variables: nhtfrq, mfilt)

• Setting environment for this exercise:

```
setenv CASENAME     ex1
setenv CESMROOT     /fis/cgd/cseg/csm/collections/cesm1_0_tutorial
setenv CASEDIR      /blhome/{$LOGNAME}/cases/$CASENAME
setenv ARCHIVEDIR   /ptmp/{$LOGNAME}/archive/$CASENAME
setenv RUNDIR       /ptmp/{$LOGNAME}/$CASENAME
```


Exercise 1

• Create a new case:

```
cd $CESMROOT/scripts
./create_newcase -case $CASEDIR \
                 -mach bluefire  \
                 -res T31_gx3v7  \
                 -compset B_1850_CN
```

• Configure the model:

```
cd $CASEDIR
./configure -case
```


Exercise 1

- Examine the contains of the directory: ***$CASEDIR/Buildconf***

```
cam.buildexe.csh        ciceconf                mct.buildlib
cam.buildnml.csh        clm.buildexe.csh        pio.buildlib
cam.input_data_list     clm.buildnml.csh        pop2.buildexe.csh
camconf                 clm.input_data_list     pop2.buildnml.csh
ccsm.buildexe.csh       clmconf                 pop2.input_data_list
cice.buildexe.csh       cpl.buildexe.csh        pop2doc
cice.buildnml.csh       cpl.buildnml.csh        sglc.buildexe.csh
cice.input_data_list    csm_share.buildlib      sglc.buildnml.csh
```

- Examine the file: ***$CASEDIR/Buildconf/cam.buildnml.csh***

```
&cam_inparm
 fincl1   = 'U'
 fincl2   = 'TREFHTMN','TREFHTMX','TREFHT','PRECC','PRECL','PSL'
 nhtfrq   = 0,-24,-24,-6,-3
 mfilt    = 1,365,30,120,240
```

This variable is added to file “h0”

Create a file “h1” with these variables

Monthly output for “h0” and daily output for “h1”

1 time sample for “h0” and 365 time samples for “h1”

Not used

Exercise 1

- Edit: ***$CASEDIR/Buildconf/cam.buildnml.csh***

```
&cam_inparm
 nhtfrq    = -24
 mfilt     = 1
 fincl2    = 'TREFHTMN','TREFHTMX','TREFHT','PRECC','PRECL','PSL'
```

Daily averages for h0

one time sample per day for h0

Delete this line

- Edit: ***$CASEDIR/env_run.xml***

cd $CASEDIR

xmlchange -file env_run.xml -id STOP_N -val '1'

xmlchange -file env_run.xml -id STOP_OPTION -val 'nmonths'

Exercise 1

- Build

cd $CASEDIR

./{$CASENAME}.bluefire.build

- Edit the run script*:* ***$CASEDIR/{$CASENAME}.bluefire.run***

#BSUB –U 37591051#4 (if available; not for auditors)

#BSUB –W 1:50

- Submit the job

cd $CASEDIR

bsub < {$CASENAME}.bluefire.run

- Check the job is running

bjobs

Exercise 1

• When job is completed, look at the output files:

cd $ARCHIVEDIR/atm/hist

ls

```
ex1.cam2.h0.0001-01-01-00000.nc  ex1.cam2.h0.0001-01-12-00000.nc  ex1.cam2.h0.0001-01-23-00000.nc
ex1.cam2.h0.0001-01-02-00000.nc  ex1.cam2.h0.0001-01-13-00000.nc  ex1.cam2.h0.0001-01-24-00000.nc
ex1.cam2.h0.0001-01-03-00000.nc  ex1.cam2.h0.0001-01-14-00000.nc  ex1.cam2.h0.0001-01-25-00000.nc
ex1.cam2.h0.0001-01-04-00000.nc  ex1.cam2.h0.0001-01-15-00000.nc  ex1.cam2.h0.0001-01-26-00000.nc
ex1.cam2.h0.0001-01-05-00000.nc  ex1.cam2.h0.0001-01-16-00000.nc  ex1.cam2.h0.0001-01-27-00000.nc
ex1.cam2.h0.0001-01-06-00000.nc  ex1.cam2.h0.0001-01-17-00000.nc  ex1.cam2.h0.0001-01-28-00000.nc
ex1.cam2.h0.0001-01-07-00000.nc  ex1.cam2.h0.0001-01-18-00000.nc  ex1.cam2.h0.0001-01-29-00000.nc
ex1.cam2.h0.0001-01-08-00000.nc  ex1.cam2.h0.0001-01-19-00000.nc  ex1.cam2.h0.0001-01-30-00000.nc
ex1.cam2.h0.0001-01-09-00000.nc  ex1.cam2.h0.0001-01-20-00000.nc  ex1.cam2.h0.0001-01-31-00000.nc
ex1.cam2.h0.0001-01-10-00000.nc  ex1.cam2.h0.0001-01-21-00000.nc  ex1.cam2.h0.0001-02-01-00000.nc
ex1.cam2.h0.0001-01-11-00000.nc  ex1.cam2.h0.0001-01-22-00000.nc
```

• Examine the output variables and the number of time samples in a history file:

cd $ARCHIVEDIR/atm/hist

ncdump -h {$CASENAME}.cam2.h0.0001-01-01-00000.nc

ncdump -v time {$CASENAME}.cam2.h0.0001-01-01-00000.nc

Exercise 2

• Using the compset: B_1850_CN, make one-month run:
On the top of the monthly history file “h0”, output:

- “h1” file with instantaneous values of T, Q, U and V every 3 hour.

- “h2” file with time-average values of T, Q, U and V every 24 hour.

Write one h1 file for every day of the month and write a single h2.
(Hint: Use namelist variables: *nhtfrq, mfilt, fincl*)

• Setting environment for this exercise:

setenv CASENAME ex2
setenv CESMROOT /fis/cgd/cseg/csm/collections/cesm1_0_tutorial
setenv CASEDIR /blhome/{$LOGNAME}/cases/$CASENAME
setenv ARCHIVEDIR /ptmp/{$LOGNAME}/archive/$CASENAME
setenv RUNDIR /ptmp/{$LOGNAME}/$CASENAME

Exercise 2

• Create a new case:

```
cd $CESMROOT/scripts
./create_newcase -case $CASEDIR \
                 -mach bluefire  \
                 -res T31_gx3v7  \
                 -compset B_1850_CN
```

• Configure the model:

```
cd $CASEDIR
./configure -case
```


Exercise 2

- Edit: ***$CASEDIR/Buildconf/cam.buildnml.csh***

```
&cam_inparm
 fincl2 = 'T:I','Q:I','U:I','V:I'
 fincl3 = 'T:A','Q:A','U:A','V:A'
 nhtfrq = 0,-3,-24
 mfilt  = 1,8,31
```

"h1" file: instantaneous values for T,Q,U,V

"h2" file: time-average values for T,Q,U,V

Output frequency: h0: monthly,
h1: 3-hour, h2: daily

Number of time samples: h0: 1 time sample
h1: 1 file per day, h2: 1 file per month

- Edit: ***$CASEDIR/env_run.xml***

cd $CASEDIR

xmlchange -file env_run.xml -id STOP_N -val '1'

xmlchange -file env_run.xml -id STOP_OPTION -val 'nmonths'

Exercise 2

- Build

cd $CASEDIR

./{$CASENAME}.bluefire.build

- Edit the run script: *$CASEDIR/{$CASENAME}.bluefire.run*

#BSUB –U 37591051#4 (if available; not for auditors)

#BSUB –W 1:50

- Submit the job

cd $CASEDIR

bsub < {$CASENAME}.bluefire.run

- Check the job is running

bjobs

Exercise 2

- When job is completed, look at the output files:

cd $ARCHIVEDIR/atm/hist

ls

ncdump –h …

"h0": monthly means for default variables

```
ex2.cam2.h0.0001-01.nc            ex2.cam2.h1.0001-01-12-00000.nc  ex2.cam2.h1.0001-01-24-00000.nc
ex2.cam2.h1.0001-01-01-00000.nc   ex2.cam2.h1.0001-01-13-00000.nc  ex2.cam2.h1.0001-01-25-00000.nc
ex2.cam2.h1.0001-01-02-00000.nc   ex2.cam2.h1.0001-01-14-00000.nc  ex2.cam2.h1.0001-01-26-00000.nc
ex2.cam2.h1.0001-01-03-00000.nc   ex2.cam2.h1.0001-01-15-00000.nc  ex2.cam2.h1.0001-01-27-00000.nc
ex2.cam2.h1.0001-01-04-00000.nc   ex2.cam2.h1.0001-01-16-00000.nc  ex2.cam2.h1.0001-01-28-00000.nc
ex2.cam2.h1.0001-01-05-00000.nc   ex2.cam2.h1.0001-01-17-00000.nc  ex2.cam2.h1.0001-01-29-00000.nc
ex2.cam2.h1.0001-01-06-00000.nc   ex2.cam2.h1.0001-01-18-00000.nc  ex2.cam2.h1.0001-01-30-00000.nc
ex2.cam2.h1.0001-01-07-00000.nc   ex2.cam2.h1.0001-01-19-00000.nc  ex2.cam2.h1.0001-01-31-00000.nc
ex2.cam2.h1.0001-01-08-00000.nc   ex2.cam2.h1.0001-01-20-00000.nc  ex2.cam2.h1.0001-02-01-00000.nc
ex2.cam2.h1.0001-01-09-00000.nc   ex2.cam2.h1.0001-01-21-00000.nc  ex2.cam2.h2.0001-01-01-00000.nc
ex2.cam2.h1.0001-01-10-00000.nc   ex2.cam2.h1.0001-01-22-00000.nc  ex2.cam2.h2.0001-02-01-00000.nc
ex2.cam2.h1.0001-01-11-00000.nc   ex2.cam2.h1.0001-01-23-00000.nc
```

"h1": 3-hour T, Q, U and V (instantaneous values)

"h2": daily means T, Q, U and V (time-average)

Exercise 3

• Using the compset: F_2000 (CAM standalone, prescribed ocn/ice, present day run)
Make three-month run: output a h1 history with your favorite set of variables over the storm track area

(Hint: Use namelist variables: *fincl2lonlat, fincl2*)

• Setting environment for this exercise:

setenv CASENAME ex3

setenv CESMROOT /fis/cgd/cseg/csm/collections/cesm1_0_tutorial

setenv CASEDIR /blhome/{$LOGNAME}/cases/$CASENAME

setenv ARCHIVEDIR /ptmp/{$LOGNAME}/archive/$CASENAME

setenv RUNDIR /ptmp/{$LOGNAME}/$CASENAME

Exercise 3

• Create a new case:

```
cd $CESMROOT/scripts
./create_newcase -case $CASEDIR \
                 -mach bluefire  \
                 -res T31_T31  \
                 -compset F_2000
```

• Configure the model:

```
cd $CASEDIR
./configure -case
```


Exercise 3

- Edit: ***$CASEDIR/Buildconf/cam.buildnml.csh***

```
&cam_inparm
   fincl2 = 'T','Q','U','V'
   fincl2lonlat = '170e:240e_30n:60n'
   nhtfrq = 0,-24
   mfilt  = 1,30
```

"h1" file: output T,Q,U,V

Select a region to output

Output frequency: h0: monthly, h1: daily

Number of time samples per files: h0: 1 sample, h1: 30 samples

- *Edit:* ***$CASEDIR/env_run.xml***

cd $CASEDIR

xmlchange -file env_run.xml -id STOP_N -val '3'

xmlchange -file env_run.xml -id STOP_OPTION -val 'nmonths'

Exercise 3

- Build

cd $CASEDIR

./{$CASENAME}.bluefire.build

- Edit the run script: *$CASEDIR/{$CASENAME}.bluefire.run*

#BSUB –U 37591051#4 (if available; not for auditors)

#BSUB –W 1:50

- Submit the job

cd $CASEDIR

bsub < {$CASENAME}.bluefire.run

- Check the job is running

bjobs

Exercise 3

• When job is completed, look at the output files:

cd $ARCHIVEDIR/atm/hist

ncdump –h $CASENAME.cam2.h1.0001-01-01-00000.nc

Notice the variables output the region you selected:

```
float T_LON_170e_to_240e_LAT_30n_to_60n(time, lev, LAT_30n_to_60n, LON_170e_to_240e) ;
        T_LON_170e_to_240e_LAT_30n_to_60n:basename = "T" ;
        T_LON_170e_to_240e_LAT_30n_to_60n:units = "K" ;
        T_LON_170e_to_240e_LAT_30n_to_60n:long_name = "Temperature" ;
        T_LON_170e_to_240e_LAT_30n_to_60n:cell_methods = "time: mean" ;
```


Exercise 4

• Using the compset: B_1850_CN, add a variable for the liquid in-cloud water path: ICLDLWP and make a 1-month run.
(Hint: Use ICLDIWP as a template for your changes.
Find the subroutine containing ICLDIWP using grepccm/findccm)

• Setting environment for this exercise:

setenv CASENAME ex4
setenv CESMROOT /fis/cgd/cseg/csm/collections/cesm1_0_tutorial
setenv CASEDIR /blhome/{$LOGNAME}/cases/$CASENAME
setenv ARCHIVEDIR /ptmp/{$LOGNAME}/archive/$CASENAME
setenv RUNDIR /ptmp/{$LOGNAME}/$CASENAME

Exercise 4

• Create a new case:

```
cd $CESMROOT/scripts
./create_newcase -case $CASEDIR \
                 -mach bluefire  \
                 -res T31_gx3v7 \
                 -compset  F_2000
```

• Configure the model:

```
cd $CASEDIR
./configure -case
```


Exercise 4

- Localize the subroutine that contains ICLDIWP by using grepccm

- First we need to build the model to create the file: Filepath

cd $CASEDIR

./{$CASENAME}.bluefire.build

- Use grepccm and findccm to localize the subroutine to modify:

cd $RUNDIR/atm/obj

/fis/cgd/home/hannay/bin/grepccm ICLDIWP

This tells us it is in: param_cldoptics.F90

- Localize param_cldoptics.F90

cd $RUNDIR/atm/obj

/fis/cgd/home/hannay/bin/findccm param_cldoptics.F90

This tells us it is in: */gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/physics/cam*

Exercise 4

• copy the subroutine ***param_cldoptics.F90*** into ***SourceMods/src.cam/***

cd $CASEDIR

cp /gpfs/proj2/fis/cgd/cseg/csm/collections/cesm1_0_tutorial/models/atm/cam/src/physics/cam/param_cldoptics.F90 SourceMods/src.cam/

• Edit: ***$CASEDIR/SourceMods/src.cam/param_cldoptics.F90***

Under the line:

```
call addfld ('ICLDIWP', 'gram/m2', pver, 'A','In-cloud ice water path',
             phys_decomp, sampling_seq='rad_lwsw')
```

Add:

```
call addfld ('ICLDLWP', 'gram/m2', pver, 'A','In-cloud liquid water path',
             phys_decomp, sampling_seq='rad_lwsw')
```

Under the line:

```
call outfld('ICLDIWP' ,cicewp , pcols,lchnk)
```

Add:

```
call outfld('ICLDIWP' ,cliqwp , pcols,lchnk)
```


Exercise 4

• Edit*: **$CASEDIR/env_run.xml***

cd $CASEDIR

xmlchange -file env_run.xml -id STOP_N -val '1'

xmlchange -file env_run.xml -id STOP_OPTION -val 'nmonths'

• Rebuild the model with the modified subroutine (should go faster !)

cd $CASEDIR

./{$CASENAME}.bluefire.build

• Edit the run script: *$CASEDIR/{$CASENAME}.bluefire.run*

#BSUB –U 37591051#4 (if available; not for auditors)

#BSUB –W 1:50

Exercise 4

- Submit the job

cd $CASEDIR

bsub < {$CASENAME}.bluefire.run

- Check the job is running

bjobs

- When job is completed, look at the output files:

cd $ARCHIVEDIR/atm/hist

ncdump -h $CASENAME.cam2.h0.0001-01.nc | grep ICLD

ncview $CASENAME.cam2.h0.0001-01.nc

Exercise 5

• Using the compset: B_1850_CN and make a 1-month run.
Change the value of Dcs (autoconversion size threshold for cloud ice to snow) to Dcs = 300.e-6_r8

• Setting environment for this exercise:

setenv CASENAME ex5
setenv CESMROOT /fis/cgd/cseg/csm/collections/cesm1_0_tutorial
setenv CASEDIR /blhome/{$LOGNAME}/cases/$CASENAME
setenv ARCHIVEDIR /ptmp/{$LOGNAME}/archive/$CASENAME
setenv RUNDIR /ptmp/{$LOGNAME}/$CASENAME

Exercise 5

• Create a new case:

```
cd $CESMROOT/scripts
./create_newcase -case $CASEDIR \
                 -mach bluefire  \
                 -res T31_gx3v7 \
                 -compset  F_2000
```

• Configure the model:

```
cd $CASEDIR
./configure -case
```


Exercise 5

• Go into CAM physics and find the subroutine that contains Dcs:

cd {$CESMROOT}/models/atm/cam/src/physics/cam

*grep Dcs **

(You can also localize the subroutine containing Dcs by using grepccm)

• copy the subroutine ***cldwat2m_micro.F90*** into ***SourceMods/src.cam/***

cd $CASEDIR

*cp {$CESMROOT}/models/atm/cam/src/physics/cam/cldwat2m_micro.F90 *

SourceMods/src.cam/

• Edit: *$CASEDIR/SourceMods/src.cam/cldwat2m_micro.F90*

Dcs = 300.e-6_r8

• Edit*:* ***$CASEDIR/env_run.xml***

cd $CASEDIR

xmlchange -file env_run.xml -id STOP_N -val '1'

xmlchange -file env_run.xml -id STOP_OPTION -val 'nmonths'

Exercise 5

- Build

cd $CASEDIR

./{$CASENAME}.bluefire.build

- Edit the run script: *$CASEDIR/{$CASENAME}.bluefire.run*

#BSUB –U 37591051#4 (if available; not for auditors)
#BSUB –W 1:50

- Submit the job

cd $CASEDIR

bsub < {$CASENAME}.bluefire.run

- Check the job is running

bjobs

- When job is completed, look at the output files:

cd $ARCHIVEDIR/atm/hist

- You can compare the files with the ones created in exercise 4