Namelist and Code 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

Part 1: Namelist Modifications

Part 2: Code Modifications

Part 3: Exercise Solutions

How to change a namelist variable ?

To understand how to change a namelist variable,
we need to understand when/how the namelists are created.

create_newcase: creates a case directory
~/$CASE that contains the files:

env_case.xml
env_conf.xml
env_build.xml
env_run.xml

configure -case: creates a sub-directory
~/$CASE/Buildconf that contains the files

cam.buildnml.csh
cice.buildnml.csh
clm.buildnml.csh
cpl.buildnml.csh
docn.buildnml.csh
sglc.buildnml.csh

$CASE.$mach.build: call $component.buildnml.csh
to create namelists (atm_in,…)
$CASE.$mach.run: in the *run directory*

atm_in
lnd_in
ice_in
ocn_in
drv_in

How to change a namelist variable ?

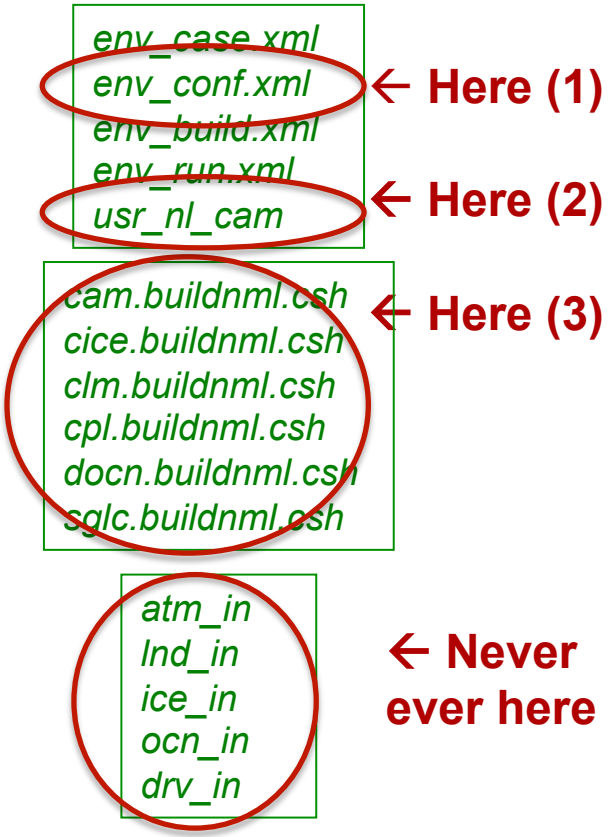
To understand how to change a namelist variable,
we need to understand when/how the namelists are created.

create_newcase: creates a case directory
~/$CASE that contains the files:

configure -case: creates a sub-directory
~/$CASE/Buildconf that contains the files

$CASE.$mach.build: call $component.buildnml.csh
to create namelists (atm_in,…)
$CASE.$mach.run: in the *run directory*

There three ways to change namelists



Let's change the output frequency in CAM**

By default, CESM outputs monthly average history files.

To change the output frequency of a CAM history file
from monthly average to daily average,
we use the namelist variable: *nhtfrq = -24*

Let's do this the "3 ways"

*** In this tutorial, most examples will be coming from the atmospheric model . Concepts are transferable to other model components.*

How to change a namelist variable ?

To understand how to change a namelist variable,
we need to understand when/how the namelists are created.

create_newcase: creates a case directory
~/$CASE that contains the files:

configure -case: creates a sub-directory
~/$CASE/Buildconf that contains the files

$CASE.$mach.build: call $component.buildnml.csh
to create namelists (atm_in,…)
$CASE.$mach.run: in the *run directory*

There three ways to change namelists

env_case.xml
env_conf.xml ← **Here (1)**
env_build.xml
env_run.xml
usr_nl_cam ← **Here (2)**

cam.buildnml.csh ← **Here (3)**
cice.buildnml.csh
clm.buildnml.csh
cpl.buildnml.csh
docn.buildnml.csh
sglc.buildnml.csh

atm_in
lnd_in
ice_in
ocn_in
drv_in

How to change a namelist variable ?

To understand how to change a namelist variable,
we need to understand when/how the namelists are created.

create_newca

configure -cas

**NB: This is a reminder:
you learned how to do this
in Christine Shields' talk**

e files

$CASE.$mach.build: call $component.buildnml.csh
to create namelists (atm_in,…)
$CASE.$mach.run: in the *run directory*

**There three ways
to change namelists**

env_case.xml
env_conf.xml ← **Here (1)**
env_build.xml
env_run.xml
usr_nl_cam ← **Here (2)**

cam.buildnml.csh ← **Here (3)**
cice.buildnml.csh
clm.buildnml.csh
cpl.buildnml.csh
docn.buildnml.csh
sglc.buildnml.csh

atm_in
lnd_in
ice_in
ocn_in
drv_in

1) Change namelists using env_conf.xml

The first method to modify a namelist variable is to edit the file env_conf.xml before configuring the model (method valid for CAM, CLM and CICE)

In env_conf.xml:

CAM_NAMELIST_OPTS: CAM namelist options that differ from default values

CLM_NAMELIST_OPTS: CLM namelist options that differ from default values

CICE_NAMELIST_OPTS: CICE namelist options for that differ from default values

For instance to change the output frequency:

xmlchange -file env_conf.xml -id CAM_NAMELIST_OPTS -val nhtfrq=-24

When you configure the model (*configure -case*), the resulting namelist variables will appear in *Buildconf/cam.buildnml.csh*

These variables CANNOT be modified once *configure -case* has been invoked without first invoking *configure -cleannamelist* or *configure -cleanall.*

How to change a namelist variable ?

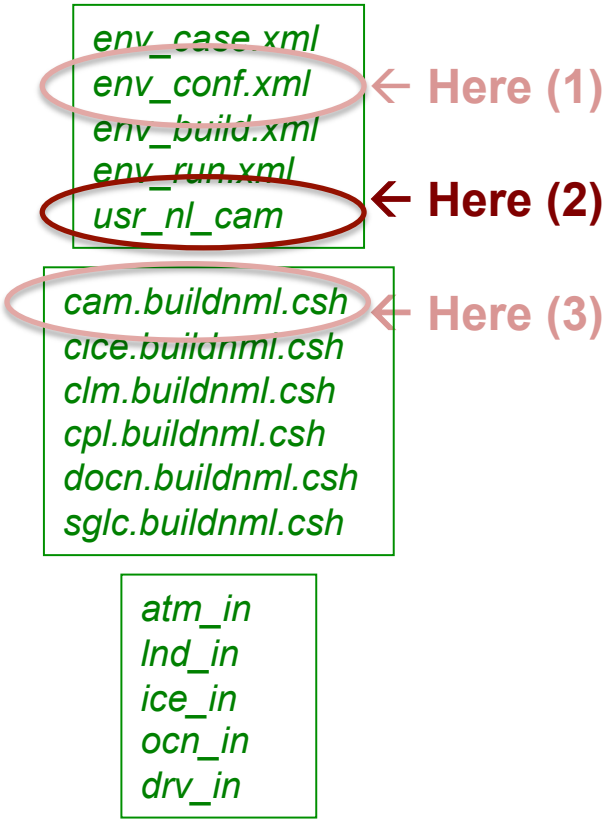
To understand how to change a namelist variable,
we need to understand when/how the namelists are created.

create_newcase: creates a case directory
~/$CASE that contains the files:

configure –case: creates a sub-directory
~/$CASE/Buildconf that contains the files

$CASE.$mach.build: call $component.buildnml.csh
to create namelists (atm_in,…)
$CASE.$mach.run: in the *run directory*

There three ways to change namelists



2) Change namelists using user_nl_cam

The second method to modify a namelist variable is to create a file *user_nl_cam* that contains the modified namelist variables for CAM.
Method also valid for CLM if creating a file *user_nl_clm* (but not valid for other components)

Using *user_nl_cam* is very useful if you need to modify numerous namelist variables.

```
&camexp
 nhtfrq=-24
/
```

The syntax is VERY important.

```
&camexp
  insert your changes (one line per change)
/
```

The file *user_nl_cam* should be placed in your case directory before you configure the model.

When you configure the model (*configure -case*), the resulting namelist variables will appear in *Buildconf/cam.buildnml.csh*

user_nl_cam CANNOT be modified once *configure -case* has been invoked without first invoking *configure -cleannamelist* or *configure -cleanall.*

How to change a namelist variable ?

To understand how to change a namelist variable,
we need to understand when/how the namelists are created.

create_newcase: creates a case directory
~/$CASE that contains the files:

configure -case: creates a sub-directory
~/$CASE/Buildconf that contains the files

$CASE.$mach.build: call $component.buildnml.csh
to create namelists (atm_in,…)
$CASE.$mach.run: in the *run directory*

There three ways to change namelists

env_case.xml
env_conf.xml ← **Here (1)**
env_build.xml
env_run.xml
usr_nl_cam ← **Here (2)**

cam.buildnml.csh ← **Here (3)**
cice.buildnml.csh
clm.buildnml.csh
cpl.buildnml.csh
docn.buildnml.csh
sglc.buildnml.csh

atm_in
lnd_in
ice_in
ocn_in
drv_in

3) Change namelists using cam.buildnml.csh

The third method to modify a namelist variable is to edit: *Buildconf/cam.buildnml.csh*

Valid for all components.

This is done after configuring the model. Useful if you forgot to include something.

You need to include your changes in the appropriate *namelist group*:

Ex: *nhtfrq* is in *cam_inparm*

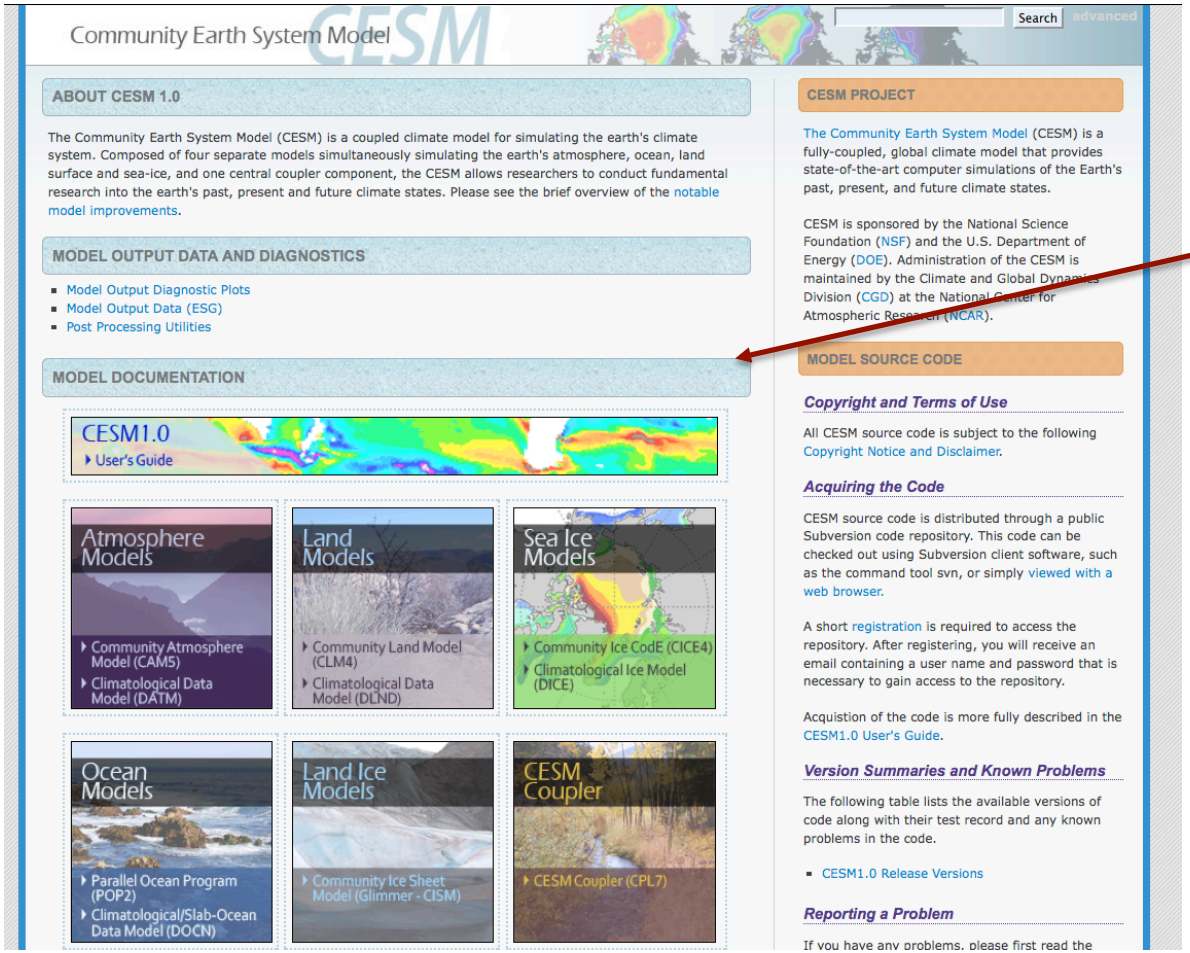
If you issue the commands:
configure -cleannamelist
or
configure -cleanall
all your changes are gone !!!

```
#! /bin/csh -f

#*****************************************************************#
#                    WARNING:                                     #
# - CAM and CLM namelist variable dtime must have same values     #
# - If the user changes any input datasets - be sure to give it a #
#   unique filename. Do not duplicate any existing input files    #
#*****************************************************************#

set exedir = /ptmp/{$LOGNAME}/case01; cd $exedir

cat >! atm_in << EOF
&aerodep_flx_nl
 aerodep_flx_datapath = '$DIN_LOC_ROOT/atm/cam/chem/trop_mozart_aero/
aero'
 aerodep_flx_file = 'aerosoldep_monthly_1849-2006_1.9x2.5_c090803.nc'
 aerodep_flx_type = 'CYCLICAL'
 aerodep_flx_ymd = 20000101
/
&cam_inparm
 nhtfrq = -24
 absems_data = '$DIN_LOC_ROOT/atm/cam/rad/
abs_ems_factors_fastvx.c030508.nc'
 bnd_topo = '$DIN_LOC_ROOT/atm/cam/topo/USGS-gtopo30_48x96_c050520.nc'
 cam_branch_file = ' '
 dtime  = 1800
 ncdata = '$DIN_LOC_ROOT/atm/cam/inic/gaus/
cami_0000-01-01_48x96_L26_c091218.nc'
 phys_loadbalance = 2
/
&chem_surfvals_nl
 ch4vmr = 1760.0e-9
 co2vmr = 367.0e-6
 f11vmr = 653.45e-12
 f12vmr = 535.0e-12
 n2ovmr = 316.0e-9
/
```

<= Include your changes here

Where to find the namelist documentation ?

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



Model documentation for each component of CESM

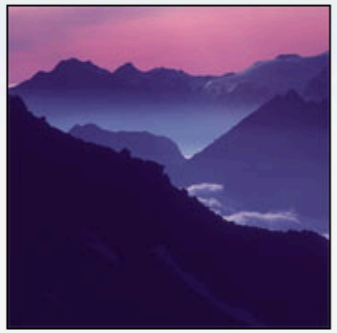
CAM namelist documentation ?

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

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 enhancements to the representation of atmospheric processes resulting in 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

CAM-5.1.1 (CESM-1.0.4)

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

CAM-5.1 (CESM-1.0.3)

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

Search Namelist Variables

CAM-5.0 (CESM-1.0 - CESM-1.0.2)

- Users Guide [html (single page)] [html (multipage)] [pdf]
- Search Namelist Variables
- Scientific description (*NCAR Tech. Note TN-486*) [ps] [pdf]

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

CAM namelist documentation ?

Search or Browse CAM Namelist Variables

This page contains the complete list of namelist variables available in CAM-5.0. 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.

nhtfrq | Search Variable Names | Show All Variable Names

AND OR (separate search terms with spaces)

Also search help text

Found 2 standard names matching query: nhtfrq

Search for nhtfrq

History and Initial Conditions Output

Namelist Variable	Type	Group
nhtfrq Array of write frequencies for each history file series. If nhtfrq(1) = 0, the file will be a monthly average. Only the first file series may be a monthly average. If nhtfrq(i) > 0, frequency is specified as number of timesteps. If nhtfrq(i) < 0, frequency is specified as number of hours. Default: 0,-24,-24,-24,-24,-24	integer(6)	cam_inparm

How to set it

type

Namelist group

CLM namelist documentation ?

http://www.cesm.ucar.edu/models/cesm1.0/clm/models/lnd/clm/doc/UsersGuide/x1997.html#nl_def

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 each namelist item and their valid values, and then we'll 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 namelists that highlight these features.

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 may change depending on the resolution, land-mask, simulation-year and other attributes. Both of these files are viewable in your web browser. Below we provide the link for them, and then expand each in turn.

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

Definition and values of namelist Variables

One set of the namelist items allows you to add fields to the output history files: `hist_fincl1`, `hist_fincl2`, `hist_fincl3`, `hist_fincl4`, `hist_fincl5`, and `hist_fincl6`. The link CLM History Fields documents all of the history fields available and gives the long-name and units for each.

CICE namelist documentation ?

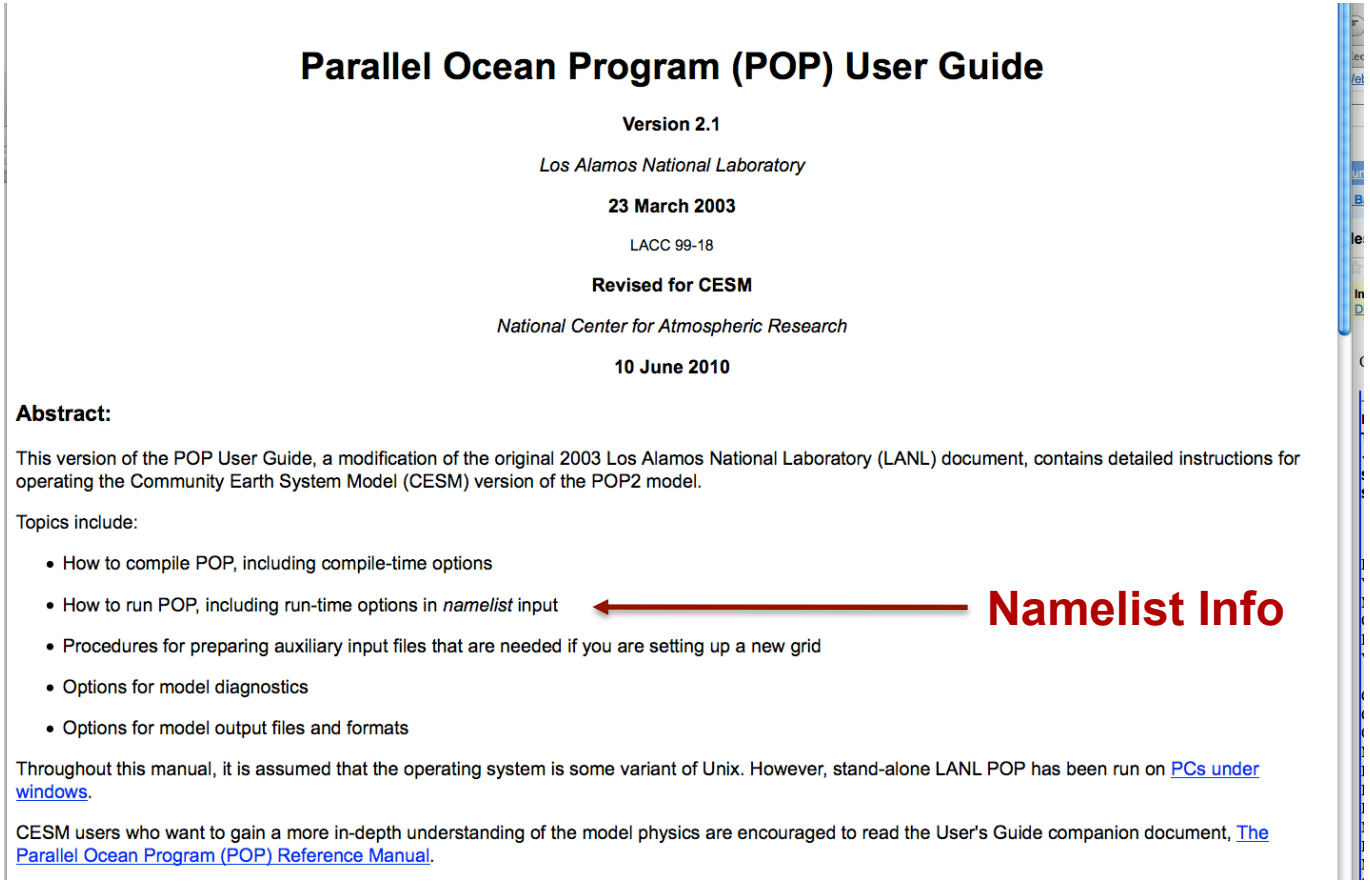
http://www.cesm.ucar.edu/models/cesm1.0/cice/doc/index.html

Contents

Section on Namelist Variables

POP 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 Info

Overview of namelist modifications

In the exercises, we will cover:

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

This can be achieved with 3 namelist variables:

- *nhtfrq*: sets the output frequency
- *fincl:* add variables to the history file
- *mfilt*: maximum 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 timesteps 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.

To control the list of fields in the history files we can use the namelist variables

h0	***h1***	***…***	***h5***
fincl1	***fincl2***	***…***	***fincl6***

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, in addition to the monthly history file “h0”, 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.

Namelist modifications: Exercises

Exercise 1

Create, configure, and build an out-of-the-box set of scripts called "case01" using the compset B_1850_CN at T31_gx3v7 resolution. Set the run length to 7 days.

Output daily averages using the variable *CAM_NAMELIST_OPTS* in env_conf.xml.

(Hint: Use namelist variables: *nhtfrq*)

Namelist modifications: Exercises

Exercise 2

Create, configure, and build an out-of-the-box set of scripts called "case02" using the compset B_1850_CN at T31_gx3v7 resolution. Set the run length to 1 month.

In addition to the monthly history file "h0", output a "h1" file with 3-hourly instantaneous values of T, Q, U and V. Make namelist variables changes by editing the file *cam.buildnml.csh*

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

Namelist modifications: Exercises

Exercise 3

Create, configure, and build an out-of-the-box set of scripts called "case03" using the compset B_1850_CN at T31_gx3v7 resolution. Set the run length to 1 month.

In addition to 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.

Make your namelist variables changes by creating a file: *user_nl_cam*

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

Part 1: Namelist Modifications

Part 2: Code Modifications

Part 3: Exercise Solutions

Your choice: The Red Pill or the Blue Pill

The Matrix (1999): Neo, the main character is offered the choice between a red pill and a blue pill.

- The blue pill would allow him to remain in the Matrix (a fictional computer-generated world)

-The red pill leading to his "escape" from the Matrix and embracing the sometimes painful truth of reality.

Analogy from Andrew Gettelman

Part II: Code modifications

This section gives an overview of simple code modifications

– Modifying a parameter in the code

– Adding an output field for variable that is not already output from the model

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

Example: Modify a parameter, Dcs

Let's modify a parameter in the CAM physics
Dcs = autoconversion size threshold for cloud ice to snow

1. Find the subroutine you want.

Go in the CESM code and look for Dcs (for instance, you can use: *grep -r Dcs **)

Dcs is in the subroutine *cldwat2m_micro.F90*

2. Copy this subroutine in SourceMods

Go your case directory and copy *cldwat2m_micro.F90* into *SourceMods/src.cam*

3. Make your modifications

Edit the value of Dcs in *SourceMods/src.cam/cldwat2m_micro.F90*

4. Compile and run the model

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

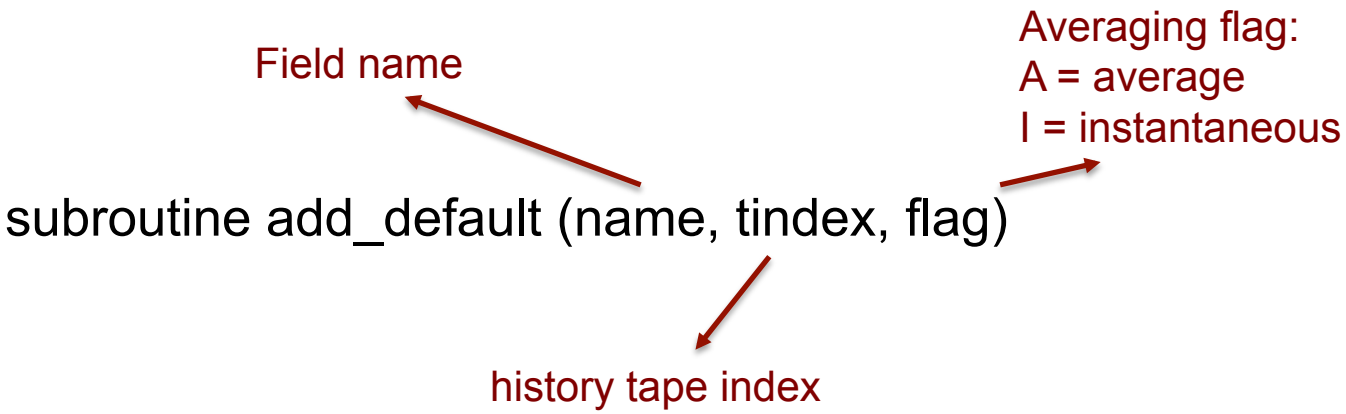
Optional (typical): indicates how often field is output. The default is every timestep. Only valid value is 'rad_lwsw' (field sampled only during LW/SW radiation calculation)

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

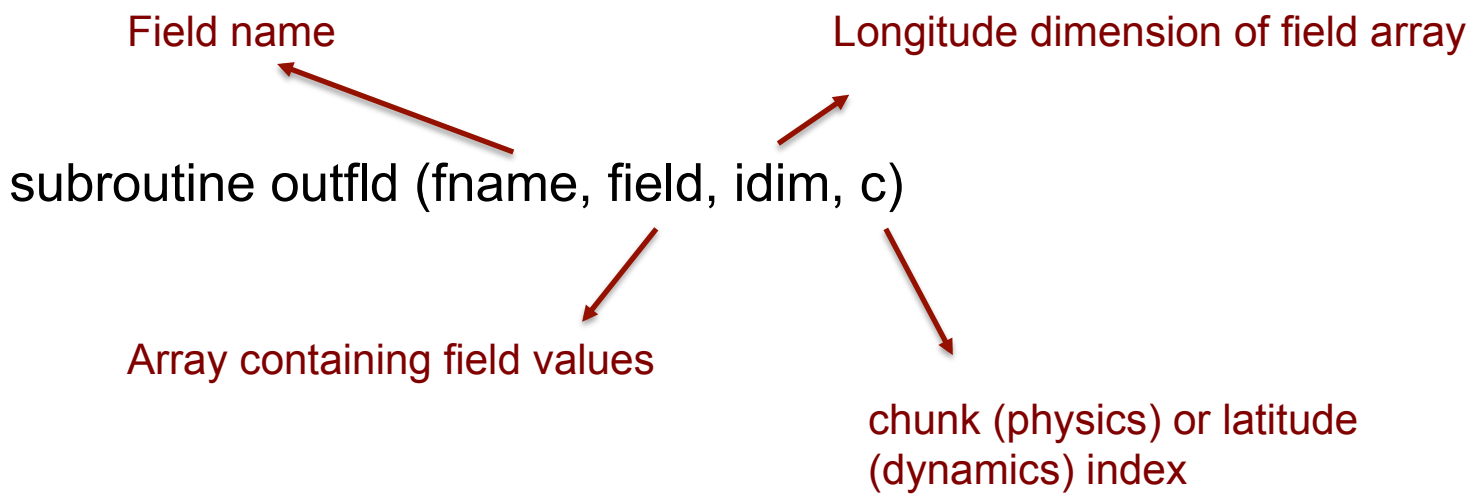


Example:

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

Syntax: outfld

outfld = 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

Create, configure, and build an out-of-the-box set of scripts called "case04" using the compset B_1850_CN at T31_gx3v7 resolution.

Add a variable called "ICLDLWP" for the liquid in-cloud water path 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

Create, configure, and build an out-of-the-box set of scripts called "case05" using the compset B_1850_CAM5_CN at T31_gx3v7 resolution.
Change the value of Dcs (autoconversion size threshold for cloud ice to snow) to Dcs = 300.e-6_r8 and make a 1-month run.

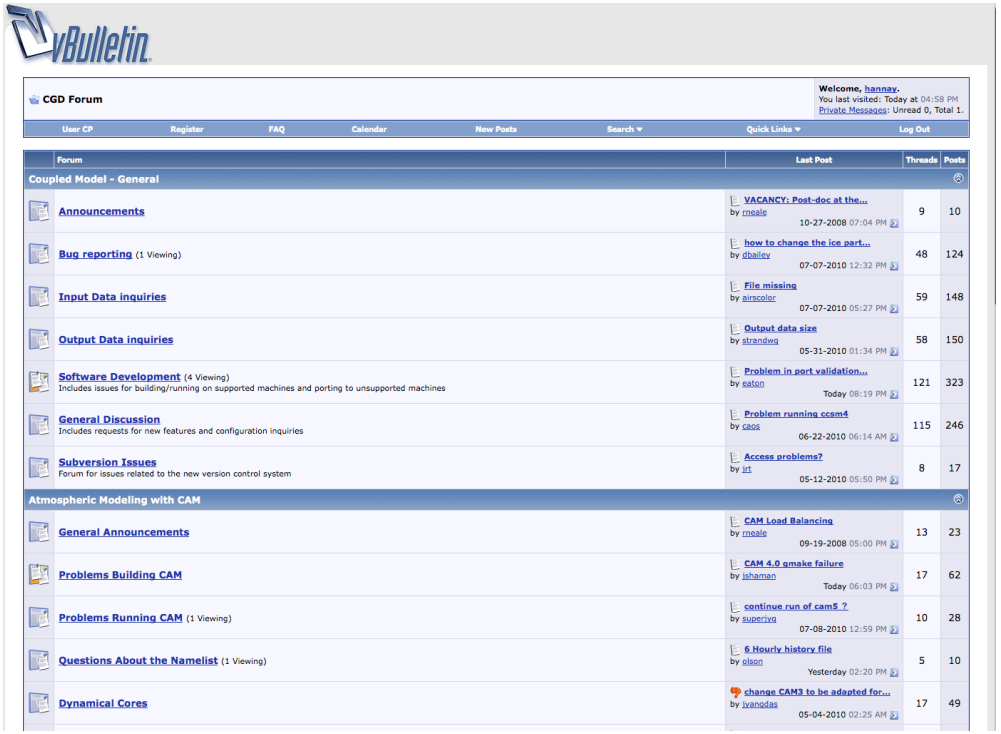
If you do more elaborate mods

- 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 you are stuck.

Document everything you do in the README.case file

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 - Solution

Exercise 1: Create, configure, and build an out-of-the-box set of scripts called “case01” using the compset B_1850_CN at T31_gx3v7 resolution. Set the run length to 7 days. Output daily averages using the variable *CAM_NAMELIST_OPTS* in env_conf.xml. (Hint: Use namelist variables: *nhtfrq*)

• Go the the scripts directory and create a new case in your home directory

```
cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/scripts
./create_newcase -case ~/case01 \
        -mach bluefire  \
        -res T31_gx3v7  \
        -compset B_1850_CN
```

• Go to the case directory and edit the file env_conf.xml to add the namelist parameter: *nhtfrq=-24*

```
cd ~/case01
./xmlchange -file env_conf.xml -id CAM_NAMELIST_OPTS -val nhtfrq=-24
```

• Configure the model:

```
cd ~/case01
./configure -case
```

• Set the run length by editing the file ***env_run.xml***

```
cd ~/case01
./xmlchange -file env_run.xml -id STOP_N -val '7'
./xmlchange  -file env_run.xml -id STOP_OPTION -val ’ndays'
```


Exercise 1 - Solution

• Build:

cd ~/case01

./case01.bluefire.build

• Edit the run script*: ~/case01/case01.bluefire.run*

#BSUB -W 1:50

• Submit the job

cd ~/case01

bsub < case01.bluefire.run

• Check the job is running

bjobs

• Look at the results:

- Examine the namelists. Find nhtfrq in the namelist. Compare with a previous run.

- When the run is completed, look at the output files in: /ptmp/$LOGNAME/archive/case01/atm/hist. Compare with a previous run. What is the output frequency of the “h0” files ?

Exercise 2 - Solution

Exercise 2: Create, configure, and build an out-of-the-box set of scripts called “case02” using the compset B_1850_CN at T31_gx3v7 resolution. Set the run length to 1 month. in addition to the monthly history file “h0”, output a “h1” file with 3-hourly instantaneous values of T, Q, U and V. Make namelist variables changes by editing the file *cam.buildnml.csh*
(Hint: Use namelist variables: *nhtfrq, fincl*)

• Go the the scripts directory and create a new case in your home directory

```
cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/scripts
./create_newcase -case ~/case02 \
        -mach bluefire  \
        -res T31_gx3v7  \
        -compset B_1850_CN
```

• Configure the model:

```
cd ~/case02
./configure -case
```

• Set the run length by editing the file ***env_run.xml***

```
cd ~/case02
xmlchange -file env_run.xml -id STOP_N -val '1’
xmlchange  -file env_run.xml -id STOP_OPTION -val ’nmonths'
```


Exercise 2 - Solution

• Edit: *~/case02/Buildconf/cam.buildnml.csh*

```
&cam_inparm
 fincl2 = 'T:I','Q:I','U:I','V:I'
 nhtfrq = 0,-3,
```

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

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

•Build:

cd ~/case02

./case02.bluefire.build

• Edit the run script*: ~/case02/case02.bluefire.run*

#BSUB -W 1:50

• Submit the job

cd ~/case02

bsub < case02.bluefire.run

• Check the job is running

bjobs

Exercise 2 - Solution

- Look at the results:

- Examine the namelists. Find nhtfrq and fincl in the namelist. Compare with case01.

- When the run is completed, look at the output files in: /ptmp/$LOGNAME/archive/case02/atm/hist. Compare with case01. What is the output frequency of the “h0” files ?

case01.cam2.h0.0001-01-01-00000.nc **-> “h0”: daily means for default variables**
case01.cam2.h0.0001-01-02-00000.nc
case01.cam2.h0.0001-01-03-00000.nc
case01.cam2.h0.0001-01-04-00000.nc
case01.cam2.h0.0001-01-05-00000.nc
case01.cam2.h0.0001-01-06-00000.nc
case01.cam2.h0.0001-01-07-00000.nc
case01.cam2.h0.0001-01-08-00000.nc

case02.cam2.h0.0001-01.nc **-> “h0”: monthly means for default variables**
case02.cam2.h1.0001-01-01-00000.nc **“h1”: 3-hour T, Q, U and V (instantaneous values)**

Exercise 3 - Solution

Exercise 3: Create, configure, and build an out-of-the-box set of scripts called "case03" using the compset B_1850_CN at T31_gx3v7 resolution. Set the run length to 1 month.
in addition to 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.

Make your namelist variables changes by creating a file: *user_nl_cam*
(Hint: Use namelist variables: *nhtfrq, mfilt, fincl*)

•Go the the scripts directory and create a new case in your home directory

```
cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/scripts
./create_newcase -case ~/case03 \
          -mach bluefire  \
          -res T31_gx3v7  \
          -compset B_1850_CN
```

• Create a file *user_nl_cam* in *~/case03*

```
&camexp
 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
/
```


Exercise 3 - Solution

- Configure the model:

cd ~/case03

./configure –case

• Look the namelist variables in *~/case03/Buildconf/cam.buildnml.csh.* Check that your changes are there:

```
&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

• Set the length of the run in: *~/case03/env_run.xml*

cd ~/case03

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

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

Exercise 3 - Solution

• Build:

cd ~/case03

./case03.bluefire.build

• Edit the run script*: ~/case03/case03.bluefire.run*

#BSUB -W 1:50

• Submit the job

cd ~/case03

bsub < case03.bluefire.run

• Check the job is running

bjobs

Exercise 3 - Solution

- Look at the results:

- Examine the namelists. Find nhtfrq, fincl, and mfilt in the namelist. Compare with case02.

- When the run is completed, look at the output files in: /ptmp/$LOGNAME/archive/case03/atm/hist. Compare with case01. What is the output frequency of the “h0” files ?

case02.cam2.h0.0001-01.nc **-> “h0”: monthly means for default variables**

case02.cam2.h1.0001-01-01-00000.nc **-> “h1”: 3-hour T, Q, U and V (instantaneous values). The file contains 249 timesteps.**

case03.cam2.h0.0001-01.nc **> “h0”: monthly means for default variables**

case03.cam2.h1.0001-01-01-00000.nc nc **-> “h1”: 3-hour T, Q, U and V (instantaneous values)**

case03.cam2.h1.0001-01-02-00000.nc **Each file contains 8 timesteps**

case03.cam2.h1.0001-01-03-00000.nc

case03.cam2.h1.0001-01-04-00000.nc

…

case03.cam2.h2.0001-01-01-00000.nc **-> “h2”: 24-hourly T, Q, U and V (mean values) The file contains 8 timesteps**

Exercise 4 - Solution

Exercise 4: Create, configure, and build an out-of-the-box set of scripts called "case04" using the compset B_1850_CN at T31_gx3v7 resolution.

Add a variable called "ICLDLWP" for the liquid in-cloud water path and make a 1-month run.
(Hint: Use ICLDIWP as a template for your changes.
Find the subroutine containing ICLDIWP using *grepccm/findccm*)

• Go the the scripts directory and create a new case in your home directory

```
cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/scripts
./create_newcase -case ~/case04 \
        -mach bluefire  \
        -res T31_gx3v7  \
        -compset B_1850_CN
```

• Configure the model:

```
cd ~/case04
configure -case
```

• Set the run length by editing the file *env_run.xml*

```
cd ~/case04
xmlchange -file env_run.xml -id STOP_N -val '1'
xmlchange  -file env_run.xml -id STOP_OPTION -val 'nmonths'
```


Exercise 4 - Solution

• Localize the subroutine that contains ICLDIWP. It is in:
/glade/proj3/cseg/collections/cesm1_0_4_tutorial/models/atm/cam/src/physics/cam/param_cldoptics.F90

• Copy this file into: *~/case04/SourceMods/src.cam*

cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/models/atm/cam/src/physics/cam/

cp param_cldoptics.F90 ~/case04/SourceMods/src.cam

• Edit: *~/case04/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('ICLDLWP' ,cliqwp , pcols,lchnk)

Exercise 4 - Solution

• Add new variable to h0 history file

Edit: *Buildconf/cam.buildnml.csh* to add

fincl1 = 'ICLDLWP'

• Build:

cd ~/case04

./case04.bluefire.build

• Edit the run script*: ~/case04/case04.bluefire.run*

#BSUB -W 1:50

• Submit the job

cd ~/case04

bsub < case04.bluefire.run

• Check the job is running

bjobs

• When the run is completed, look at the results: */ptmp/$LOGNAME/archive/case04/atm/hist*

Check that the variable ICLDLWP is in the h0 file>

ncdump -h case04.cam2.h0.0001-01.nc | grep ICLDLWP

Exercise 5 - Solution

Exercise 5: Create, configure, and build an out-of-the-box set of scripts called “case05” using the compset B_1850_CN at T31_gx3v7 resolution. Change the value of Dcs (autoconversion size threshold for cloud ice to snow) to Dcs = 300.e-6_r8 and make a 1-month run.

• Create a new case:

```
cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/scripts
./create_newcase -case ~/case05 \
        -mach bluefire  \
        -res T31_gx3v7 \
        -compset  B_1850_CAM5_CN
```

• Configure the model:

```
cd ~/case05
./configure -case
```

• Set the run length by editing the file *env_run.xml*

```
cd ~/case05
xmlchange -file env_run.xml -id STOP_N -val '1’
xmlchange  -file env_run.xml -id STOP_OPTION -val ’nmonths’
```


Exercise 5 - Solution

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

cd /glade/proj3/cseg/collections/cesm1_0_4_tutorial/models/atm/cam/src/physics/cam

*grep Dcs **

• copy the subroutine *cldwat2m_micro.F90* into *~/case05/SourceMods/src.cam/*

cd ~/case05

*cp /glade/proj3/cseg/collections/cesm1_0_4_tutorial/models/atm/cam/src/physics/cam/cldwat2m_micro.F90 *
SourceMods/src.cam/

• Edit: *~/case05/SourceMods/src.cam/cldwat2m_micro.F90*

Dcs = 300.e-6_r8

Exercise 5 - Solution

• Build:

cd ~/case05

./case05.bluefire.build

• Edit the run script*: ~/case05/case05.bluefire.run*

#BSUB -W 1:50

• Submit the job

cd ~/case05

bsub < case05.bluefire.run

• Check the job is running

bjobs

• Look at the results: