

# **CESM Research Tools: CLM4 in CESM1.0.4 User's Guide Documentation**

**Erik Kluzek**

NCAR

Mar-31-2012

\$URL: https://svn-ccsm-

models.cgd.ucar.edu/clm2/branch\_tags/cesm1\_0\_4\_rel\_tags/cesm1\_0\_4\_n05\_clm4\_0\_32/model

\$

## **CESM Research Tools: CLM4 in CESM1.0.4 User's Guide Documentation**

by Erik Kluzek

The user's guide to CLM4 in CESM1.0.4 which is the active land surface model component of CESM1.0.4. The purpose of this guide is to instruct both the novice and experienced user, as well as CLM developers in the use of CLM4 for land-surface climate modeling.

# **Dedication**

Dedicated to the Land Model Working Group, winners of the 2008 CCSM Distinguished Achievement Award. May you continue to collaborate together well, and continue to drive the science of land surface modeling forward with your diligent and persistent efforts.

# Table of Contents

Acknowledgments .....	x
Introduction .....	xi
Introduction to the CLM4 User's Guide .....	xii
Important Notes and Best Practices for Usage of CLM4.....	xiv
How to Use This Document.....	xv
What is new with CLM4 in CESM1.0.4 since previous public releases?.....	xvi
What is new with CLM4 in CESM1.0.4 since the June 15th, 2011 CESM1.0.3 release?.....	xvi
What is new with CLM4 in CESM1.0.4 Science since CLM4 in CESM1.0.3? .....	xvi
What is new with CLM4 in CESM1.0.4 Software since CLM4 in CESM1.0.3? .....	xvi
What was new with CLM4 in CESM1.0.3 since the December 8th, 2010 CESM1.0.2 release? ...	xvi
What was new with CLM4 in CESM1.0.3 Science since CLM4 in CESM1.0.2?.....	xvii
What is new with CLM4 in CESM1.0.3 Software since CLM4 in CESM1.0.2? .....	xvii
What was new with CLM4 in CESM1.0.2 (in CESM1.0.2) since the September 17th, 2010 CESM1.0.1 release? .....	xviii
What was new with CLM4 in CESM1.0.2 Science since CLM4 in CESM1.0.1?.....	xix
What was new with CLM4 in CESM1.0.2 Software since CLM4 in CESM1.0.1?.....	xix
What was new with CLM4 in CESM1.0.1 (in CESM1.0.1) since the April 1st, 2010 CCSM4.0 release? .....	xix
What was new with CLM4 in CESM1.0.1 Science since CCSM4.0? .....	xx
What was new with CLM4 in CESM1.0.1 Software since CCSM4.0? .....	xx
What was new with CLM4.0.00 since CLM3.5?.....	xxi
What was new with CLM4.0.00 Science?.....	xxi
Biogeophysics and Hydrology .....	xxi
Snow Model .....	xxii
Surface Datasets .....	xxii
Biogeochemistry .....	xxii
Miscellaneous Changes.....	xxiii
Summary of Science Changes .....	xxiii
What is new with CLM4.0.00 Software Infrastructure? .....	xxiii
What are The New Configuration Options in CLM4.0.00? .....	xxiv
What are The New Namelist Options in CLM4.0.00? .....	xxiv
What are The New History Fields? .....	xxv
Quickstart to using CLM4 .....	xxxiii
What is scientifically validated and functional in CLM4? .....	xxxv
Standard Configuration and Namelist Options that are Validated .....	xxxv
Configure Modes NOT scientifically validated, documented, supported or, in some cases, even advised to be used:.....	xxxv
Namelist options that should NOT be exercised: .....	xxxvi
Build-Namelist options that should NOT be exercised: .....	xxxvi
Namelist items that should NOT be exercised: .....	xxxvi

<b>What are the UNIX utilities required to use CLM? .....</b>	<b>xxxviii</b>
<b>Other resources to get help from .....</b>	<b>xxxix</b>
The CESM User's-Guide .....	xxxix
The CESM Bulletin Board.....	xxxix
The CLM web pages .....	xxxix
Reporting bugs in CLM4 .....	xl
<b>1. How to customize the configuration for a case with CLM.....</b>	<b>1</b>
Choosing a compset using CLM .....	1
Compsets coupled to data atmosphere and stub ocean/sea-ice ("I" compsets) .....	1
Compsets coupled to active atmosphere with data ocean.....	3
Fully coupled compsets with fully active ocean, sea-ice, and atmosphere .....	3
Conclusion to choosing a compset .....	3
Customizing the CLM configuration .....	3
CLM Script configuration items.....	4
Configure time User Namelist .....	9
Precedence of Options .....	10
Setting Your Initial Conditions File.....	10
Doing a hybrid simulation to provide initial conditions .....	11
Doing a branch simulation to provide initial conditions.....	11
Providing a finidat file in your <code>user_nl_clm</code> file .....	11
Adding a finidat file to the XML database.....	12
Other noteworthy configuration items .....	12
Downloading DATM Forcing Data .....	14
Customizing via the template files .....	14
More information on the CLM <b>configure</b> script .....	15
Help on CLM <b>configure</b> .....	15
Customizing the CLM namelist .....	15
Definition of Namelist items and their default values .....	15
Definition of CLM history variables .....	16
Examples of using different namelist features .....	60
The default namelist.....	60
Adding/removing fields on your primary history file .....	61
Adding auxiliary history files and changing output frequency .....	61
Removing all history fields .....	62
Various ways to change history output averaging flags .....	62
Outputting history files as a vector in order to analyze the plant function types within gridcells .....	63
Conclusion to namelist examples.....	64
Customizing the DATM Namelist and Streams files .....	64
<code>CLM_QIAN</code> mode and it's DATM settings .....	66
<code>CLM1PT</code> mode and it's DATM settings.....	67
<code>CPLHIST3HrWx</code> mode and it's DATM settings .....	67
Conclusion to customizing chapter .....	68

<b>2. Using the CLM tools to create your own input datasets .....</b>	<b>69</b>
Common environment variables and options used in building the FORTRAN tools .....	69
General information on running the FORTRAN tools.....	73
Running FORTRAN tools with namelists.....	73
Running FORTRAN tools with command line options .....	73
Running FORTRAN tools built with SMP=TRUE .....	74
The File Creation Process .....	74
Using the <b>cprnc</b> tool to compare two history files.....	76
Using <b>interpinic</b> to interpolate initial conditions to different resolutions .....	78
Using <b>mkgriddata</b> to create grid datasets .....	79
mkgriddata namelist .....	80
Convert CESM domain files to CLM grid files .....	82
Create single point or regional area grid files.....	82
Convert CAM files to CLM grid files (deprecated).....	83
Using <b>mkdatadomain</b> to create domain datasets for DATM or docn from CLM grid datasets ....	83
Using <b>mksurfdata</b> to create surface datasets from grid datasets .....	84
Running <b>mksurfdata.pl</b> .....	84
Running <b>mksurfdata</b> by Hand.....	85
Preparing your <b>mksurfdata</b> namelist .....	86
Experimental options to <b>mksurfdata</b> .....	88
Standard Practices when using <b>mksurfdata</b> .....	88
Using NCL scripts <b>ndepregrid.ncl</b> and <b>aerdepregrid.ncl</b> to interpolate aerosol deposition datasets	
89	
Using <b>ndepregrid.ncl</b> to interpolate Nitrogen deposition datasets.....	90
Using <b>aerdepregrid.ncl</b> to interpolate Aerosol deposition datasets.....	91
How to Customize Datasets for particular Observational Sites .....	91
Conclusion of tools description.....	92
<b>3. Adding New Resolutions or New Files to the build-namelist Database.....</b>	<b>93</b>
Managing Your Own Data-files .....	93
Adding Resolution Names .....	94
Adding or Changing Default Filenames .....	95
What are the required files?.....	96
<b>4. How to run some special cases .....</b>	<b>99</b>
Running with the prognostic crop model on.....	99
Running with the irrigation model on .....	100
Spinning up the Satellite Phenology Model (CLMSP spinup) .....	100
Spinning up the biogeochemistry Carbon-Nitrogen Model (CN spinup) .....	101
Spinning up the Carbon-Nitrogen Dynamic Global Vegetation Model (CNDV spinup) .....	102
Running with MOAR data as atmospheric forcing to spinup the model .....	103
Running with your own previous simulation as atmospheric forcing to spinup the model .....	104
Doing perturbation error growth tests .....	106
Running stand-alone CLM with transient historical CO <sub>2</sub> concentration.....	112

<b>5. How to run Single-Point/Regional cases .....</b>	<b>116</b>
Which Single Point Option Should I choose?.....	116
Running PTS_MODE configurations .....	117
Warning about Running with a Single-Processor on a Batch Machine .....	118
Running Supported Single-point/Regional Datasets.....	119
Running Supported Single-point Datasets that have their own Atmospheric Forcing .....	120
Creating your own single-point/regional surface datasets .....	122
Using getregional_datasets.pl to get a complete suite of single-point/regional surface datasets from global ones .....	125
Running with your own atmosphere forcing.....	127
<b>6. How to run PTCLM1.110726 .....</b>	<b>133</b>
Introduction to PTCLM .....	133
Guide to the options of PTCLM .....	134
Overview on using PTCLM .....	135
Details on the options of PTCLM .....	137
Examples using PTCLM.....	141
Adding data for use by PTCLM.....	143
PTCLM Group Site Lists.....	143
Dynamic Land-Use Change Files for use by PTCLM .....	144
Converting AmeriFlux Data for use by PTCLM .....	144
PTCLM transient example over a shorter time period .....	145
A bit about the structure of PTCLM, what it does, and how it works .....	146
<b>7. Trouble Shooting Problems.....</b>	<b>147</b>
Trouble with Configuration.....	147
Trouble with Building .....	148
Trouble with Running .....	149
Tracking Problems by Querying Log Files .....	149
The coupler log file .....	149
The ccsm log file .....	150
The CLM log file .....	152
The DATM log file .....	152
The batch log files .....	153
General Advice on Debugging Run time Problems .....	153
Run in DEBUG mode .....	153
Run with a smaller set of processors.....	154
Run in serial mode with a single processor .....	154
Run at a lower resolution .....	155
Run a simpler case .....	155
Run with a debugger .....	155
<b>A. Editing Template Files Before Configure.....</b>	<b>156</b>
Outline of the CLM template .....	156
Outline of the DATM template.....	157
Adding a new DATM_MODE to the DATM template .....	158

<b>B. Using the Script <code>runinit_ibm.csh</code> to both Run CLM and Interpolate Datasets .....</b>	<b>161</b>
<b>C. Scripts for testing CLM.....</b>	<b>162</b>
Testing CLM Using the CESM Test Scripts .....	162
Testing CLM Using the CLM Stand-Alone Testing Scripts .....	162
Testing PTCLM use the PTCLM test script.....	163
<b>D. Building the Users-Guide Documentation for CLM.....</b>	<b>164</b>

# List of Examples

1-1. Example <code>user_nl_clm</code> namelist file .....	9
1-2. Default CLM Namelist.....	60
1-3. Example <code>user_nl_clm</code> namelist adding and removing fields on primary history file .....	61
1-4. Example <code>user_nl_clm</code> namelist adding auxiliary history files and changing output frequency .....	61
1-5. Example <code>user_nl_clm</code> namelist removing all history fields .....	62
1-6. Example <code>user_nl_clm</code> namelist with various ways to average history fields .....	63
1-7. Example <code>user_nl_clm</code> namelist outputting some files in 1D Vector format.....	64
2-1. Example of running CLM to create a template file for <b>interpnic</b> to interpolate to .....	78
2-2. Example of building and running <b>interpnic</b> to interpolate a 1-degree <code>finidat</code> dataset to 10x15 ..	79
2-3. Example <b>mkgriddata</b> namelist to convert CESM 4x5 domain files to CLM grid files.....	82
2-4. Example <b>mkgriddata</b> namelist to create regional grid over Amazon .....	83
2-5. Example <b>mkdatadomain</b> namelist to create a domain file from CLM frac and grid data files .....	84
2-6. Example of running <b>mksurfdata.pl</b> to create a 4x5 resolution <code>fsurdat</code> for a 1850 simulation year 85	
2-7. Getting the raw datasets for <b>mksurfdata</b> to your local machine using the <b>check_input_data</b> script	87
4-1. Example Crop Simulation .....	99
4-2. Example Irrigation Simulation .....	100
4-3. Example AD_SPINUP Simulation.....	101
4-4. Example EXIT_SPINUP Simulation .....	101
4-5. Example Final CN Spinup Simulation .....	102
4-6. Example CNDV Spinup Simulation.....	103
4-7. Example Simulation with MOAR Data on bluefire .....	104
4-8. Example Fully Coupled Simulation to Create Data to Force Next Example Simulation .....	105
4-9. Example Simulation Forced with Data from the Previous Simulation .....	105
4-10. Example non-perturbed error growth simulation .....	107
4-11. Example perturbed error growth simulation.....	107
4-12. Example Transient Simulation with Historical CO <sub>2</sub> .....	113
5-1. Example of running CLM over a single-point test site in Brazil with the default Qian atmosphere data forcing. .....	119
5-2. Example of running CLM over the single-point of Mexicocity Mexico with the default Qian atmosphere data forcing. ....	119
5-3. Example of running CLM over the single-point of Vancouver Canada with supplied atmospheric forcing data for Vancouver.....	121
5-4. Example of running <b>getregional_datasets.pl</b> to get datasets for a specific region over Alaska .....	126
5-5. Example of using CLM_USRDAT_NAME to run a simulation using user datasets for a specific region over Alaska .....	127
5-6. Example of setting up a case with your own atmosphere forcing .....	128
5-7. Example of DATM streams files with your own forcing for 3-hourly data .....	129
6-1. Example of running PTCLM1 for US-UMB on bluefire .....	133
6-2. Example of running PTCLM for the Mexicocity supported single point dataset .....	141
6-3. Example of running PTCLM for a spinup simulation with Qian data for tower years.....	142
6-4. Example of running PTCLM on a generic machine with global PFT and soil types dataset .....	142
6-5. Example of running PTCLM for transient land-use 1991-2006 for US-Ha1 on bluefire .....	145
7-1. Example of configure problem with missing datasets.....	147
A-1. Getting help with <b>build_streams</b> for DATM .....	159

# Acknowledgments

I want to acknowledge all of the people that helped review or edit the model documentation: David Lawrence, Samuel Levis, Keith Oleson, and Sean Swenson. Thank you for your help in catching errors, and making the document more understandable and readable. Our readers thank you as well, as now it is much easier for them to digest. Any mistakes, or errors are all mine. If you run across one of those errors, please let us know, by following the Section called *Reporting bugs in CLM4* in *Other resources to get help from*. I also want to thank Sheri Mickelson, for her work in doing perturbation analysis on bluefire and intrepid, which was used in our initial versions of this User's Guide. We also want to thank the original authors of PTCLM: Daniel M. Ricciuto, Dali Wang, Peter E. Thornton, Wilfred M. Post, and R. Quinn Thomas for providing a nice addition to the CESM effort. We also want to thank the folks at University of Michigan Biological Stations (US-UMB) who allowed us to use their Fluxnet station data and import it into our inputdata repository, especially Gil Bohrer the PI on record for this site (see the Section called *Converting AmeriFlux Data for use by PTCLM* in Chapter 6 for permission information on using this data).

# Introduction

The Community Land Model (CLM4 in CESM1.0.4) is the latest in a series of global land models developed by the CESM Land Model Working Group (LMWG) and maintained at the National Center for Atmospheric Research (NCAR). This guide is intended to instruct both the novice and experienced user on running CLM. This guide pertains to the latest version CLM4 in CESM1.0.4 available for download from the public release subversion repository as a part of CESM1.0.4. Documentation may be different if you are using an older version, you should either update to the latest version, or use the documentation inside your own source tree. There is information in the `ChangeLog` file and in the *What is new with CLM4 in CESM1.0.4 since previous public releases?* regarding the changes from previous versions of CESM.

The novice user should read Chapter 1 in detail before beginning work, while the expert user should read *What is new with CLM4 in CESM1.0.4 since previous public releases?* and *Quickstart to using CLM4* chapters, and then use the more detailed chapters as reference. Before novice users go onto more technical problems covered in Chapter 2, Chapter 3, Chapter 4, or Chapter 5 they should know the material covered in Chapter 1 and be able to replicate some of the examples given there.

All users should read the *How to Use This Document* and *Other resources to get help from* sections to understand the document conventions and the various ways of getting help on using CLM4. Users should also read the *What is scientifically validated and functional in CLM4?* section to see if their planned use of the model is something that has been scientifically validated and well tested. Users that are NOT using NCAR machines or our list of well tested machines should also read the *What are the UNIX utilities required to use CLM?* section to make sure they have all the required UNIX utilities on the system they want to do their work.

# Introduction to the CLM4 User's Guide

## **What is in here anyway?**

Here in the introduction we first give a simple guide to understand the document conventions in *How to Use This Document*. The next section *What is new with CLM4 in CESM1.0.4 since previous public releases?* describes the differences between CLM4 in CESM1.0.4 and CLM4.0.00 (for each CESM release version up to CESM1.0.4) as well as between CLM4.0.00 and CLM3.5, both from a scientific as well as a software engineering point of view. It also talks about differences in the configuration, namelist, and history fields. The next section *Quickstart to using CLM4* is for users that are already experts in using CLM and gives a quickstart guide to the bare details on how to use CLM4. The next *What is scientifically validated and functional in CLM4?* tells you about what has been extensively tested and scientifically validated (and maybe more importantly) what has NOT. *What are the UNIX utilities required to use CLM?* lists the UNIX utilities required to use CLM4 and is important if you are running on non-NCAR machines, generic local machines, or machines NOT as well tested by us at NCAR. Next we have *Important Notes and Best Practices for Usage of CLM4* to detail some of the best practices for using CLM4 for science. The last introductory section is *Other resources to get help from* which lists different resources for getting help with CESM1.0 and CLM4.

Chapter 1 goes into detail on how to setup and run simulations with CLM4 and especially how to customize cases. Details of **configure** modes and **build-namelist** options as well as namelist options are given in this chapter.

Chapter 2 gives instructions on the CLM4 tools for creating input datasets for use by CLM, for the expert user. There's an overview of what each tool does, and some general notes on how to build the FORTRAN tools. Then each tool is described in detail along with different ways in which the tool might be used. A final section on how to customize datasets for observational sites for very savvy expert users is given as the last section of this chapter.

As a followup to the tools chapter, Chapter 3 tells how to add files to the XML database for **build-namelist** to use. This is important if you want to use the XML database to automatically select user-created input files that you have created when you setup new cases with CLM.

In Chapter 4, again for the expert user, we give details on how to do some particularly difficult special cases. For example, we give the protocol for spinning up both the CLMCN model and CLM with dynamic vegetation active (CNDV). We give instructions to do a spinup case from a previous case with Coupler history output for atmospheric forcing. We also give instructions on running both the prognostic crop and irrigation models. We also review how to validate a port to a new machine using the Perturbation error growth technique. Lastly we tell the user how to use the DATM model to send historical CO<sub>2</sub> data to CLM.

Chapter 5 outlines how to do single-point or regional simulations using CLM4. This is useful to either compare CLM simulations with point observational stations, such as tower sites (which might include your own atmospheric forcing), or to do quick simulations with CLM for example to test a new parameterization. There are several different ways given on how to perform single-point simulations which range from simple PTS\_MODE to more complex where you create all your own datasets, tying into Chapter 2 and also Chapter 3 to add the files into the **build-namelist** XML database. After this chapter Chapter 6 chapter outlines how to use the PTCLM python script to help you run single-point simulations.

Finally, Chapter 7 gives some guidance on trouble-shooting problems when using CLM4. It doesn't

cover all possible problems with CLM, but gives you some guidelines for things that can be done for some common problems.

In the appendices we talk about some issues that are useful for advanced users and developers of CLM. In Appendix A we give some basic background to the CLM developer on how to edit the `models/lnd/clm/bld/clm.cp17.template`. This is a very difficult exercise and we don't recommend it for any, but the most advanced users of CLM who are also experts in UNIX and UNIX scripting.

In Appendix B we go over how to run the script `runinit_ibm.csh`" that will interpolate standard resolution initial condition dataset to several other resolutions at once. It also runs CLM to create template files as well as doing the interpolation using `interpinic`. In general this is only something that a developer would want to do. Most users will only want to interpolate for a few specific resolutions.

In Appendix C we go over the automated testing scripts for validating that the CLM is working correctly. The test scripts run many different configurations and options with CLM making sure that they work, as well as doing automated testing to verify restarts are working correctly, and testing at many different resolutions. In general this is an activity important only for a developer of CLM, but could also be used by users who are doing extensive code modifications and want to ensure that the model continues to work correctly.

Finally in Appendix D we give instructions on how to build the documentation associated with CLM (i.e. how to build this document). This document is included in every CLM distribution and can be built so that you can view a local copy rather than having to go to the CESM website. This also could be useful for developers who need to update the documentation due to changes they have made.

# Important Notes and Best Practices for Usage of CLM4

- When running with CN, it is critical to begin with initial conditions that are provided with the release or to spin the model up following the CN spinup procedure before conducting scientific runs (see the Section called *Spinning up the biogeochemistry Carbon-Nitrogen Model (CN spinup)* in Chapter 4. Simulations without a proper spinup will effectively be starting from an unvegetated world. See the Section called *Setting Your Initial Conditions File* in Chapter 1 for information on how to provide initial conditions for your simulation.
- Initial condition files are provided for fully coupled BCN and offline ICN cases for 1850 and 2000 at 1deg, 2deg, and T31 resolutions. There's also an initial condition file for ICN with the prognostic crop model for 2000 at 2deg resolution, and one with CLMSP for 2000 at 2deg resolution. We also have initial conditions for offline CNDV for 1850. And there are interpolated datasets for 4x5 and 10x15 resolution for 1850. The 1850 initial condition files are in 'reasonable' equilibrium. The 2000 initial condition files represent the model state for the year 2000, and have been taken from transient simulations. Therefore, by design the year 2000 initial condition files do not represent an equilibrium state. Note also that spinning the 2000 initial conditions out to equilibrium will not reflect the best estimate of the real carbon/nitrogen state for the year 2000.
- Users can generate initial condition files at different resolutions by using the CLM tool **interpinic** to interpolate from one of the provided resolutions to the resolution of interest. Interpolated initial condition files may no longer be in 'reasonable' equilibrium.
- Aerosol deposition is a required field to CLM4 sent from the atmosphere model. Simulations without aerosol deposition will exhibit unreasonably high snow albedos. The model sends aerosol deposition from the atmospheric model (either CAM or DATM). When running with prescribed aerosol the atmosphere model will interpolate the aerosols from 2-degree resolution to the resolution the atmosphere model is running at.

# How to Use This Document

## ***Conventions used in the document for code and commands***

This section provides the details in using CLM with the CESM modeling system. Links to descriptions and definitions have been provided in the code below. We use the same conventions used in the CESM documentation as outlined below.

Throughout the document this style is used to indicate shell commands and options, fragments of code, namelist variables, etc. Where examples from an interactive shell session are presented, lines starting with > indicate the shell prompt. A backslash "\\" at the end of a line means the line continues onto the next one (as it does in standard UNIX shell). Note that \$EDITOR" is used to refer to the text editor of your choice. \$EDITOR is a standard UNIX environment variable and should be set on most UNIX systems. Comment lines are signaled with a "#" sign, which is the standard UNIX comment sign as well. \$CSMDATA is used to denote the path to the inputdata directory for your CESM data.

```
> This is a shell prompt with commands \
that continues to the following line.
> $EDITOR filename # means you are using a text editor to edit "filename"
# This is a comment line
```

# **What is new with CLM4 in CESM1.0.4 since previous public releases?**

In this section we list the updates that have occurred to CLM4 since previous public releases. In the first sections we describe changes in CLM4 in CESM1.0.4 since the CCSM4.0 release, and in the last one we describe changes from CLM3.5 to CLM4.0.00 release. Note, that the changes in the last section do NOT include the more recent changes given in the first section, but only list the changes from CLM3.5 to the CLM4.0.00 release that was part of the CCSM4.0 public release. We will describe both the changes in the science in the model as the software engineering changes. Software engineering changes includes the configure and namelist changes, as well as the new history fields.

## **What is new with CLM4 in CESM1.0.4 since the June 15th, 2011 CESM1.0.3 release?**

### **What is new with CLM4 in CESM1.0.4 Science since CLM4 in CESM1.0.3?**

No new science was introduced in CLM4 in CESM1.0.4. The scientific model version is the same as the CESM1.0.3 release.

### **What is new with CLM4 in CESM1.0.4 Software since CLM4 in CESM1.0.3?**

Several bug fixes for CESM1.0.3 were implemented. A couple issues with CNDV were resolved. A issue with the **configure** step where it would hang with a bad `user_nl_clm` file was fixed. A few issues with history variables was fixed. We fixed it so that you could build with C13 on bluefire. Domain files and fraction files were made to be consistent. A problem with Gregorian input was fixed. Several issues with PTCLM were fixed. Monthly average files are no longer required to be one per month. CLM1PT mode was changed to "cycle" mode by default so that simulations running beyond the valid data range would NOT be invalidated. And a couple problems with PIO were fixed. The I1850SPINUPCN was changed to point to an appropriate 1850 case rather than a transient case.

For history variable issues: the variables: CISUN, CISHA, ALPHAPSNSUN, and ALPHAPSNSHA were fixed, and the units were corrected for: H2OSNO\_TOP, HC, and HCSOI.

The domain and fraction files that were made to be consistent were for the following resolutions: T85\_T85, T42\_T42, T31\_T31, f10\_f10, f45\_45, f19\_f19, f09\_f09, f05\_f05, and f02\_f02.

The PTCLM issues fixed were as follows. The version of PTCLM1 was updated from PTCLM1.110504 to PTCLM1.110726. The soil data was corrected for the US-UMB site. The final spinup time was fixed. And a problem with transient compsets was fixed.

## **What was new with CLM4 in CESM1.0.3 since the December 8th, 2010 CESM1.0.2 release?**

### **What was new with CLM4 in CESM1.0.3 Science since CLM4 in CESM1.0.2?**

A prognostic crop model option was added in (based on Agro-IBIS) from work by Samuel Levis. The crop model adds in four new vegetation types for: soybean, winter and spring temperate cereals, and corn on their own separate columns. Winter cereal was added as a PFT type, but doesn't exist in the input datasets, only spring cereal is used. Winter cereal also has NOT been scientifically validated or tested. The model manages these by modeling both planting and harvesting. See the Section called *Running with the prognostic crop model on* in Chapter 4 for an example of running with it.

Also an irrigation model was added from work by Samuel Levis and Bill Sacks. This model takes water from runoff and adds it to the generic crop for those areas equipped for irrigation. See the Section called *Running with the irrigation model on* in Chapter 4 for an example of running with it. Please note that the irrigation model does not currently work with the new crop model described above, but rather only works with the generic crop

### **What is new with CLM4 in CESM1.0.3 Software since CLM4 in CESM1.0.2?**

Since CLM4 in CESM1.0.2 all Input/Output uses PIO (Parallel Input/Output package). Restart history files are now NetCDF. Input and output files can be read/written in parallel using PIO. We removed a list of old CPP defines and removed the old misc/preproc.h files. Also a new tool for working with single-point sites was added into the CESM scripts the Python tool PTCLM. We have a complete Chapter 6 chapter on it's use.

New configuration options:

-crop  
-noio

Configuration options removed:

-dust  
-progsslt

New build-namelist options:

-irrig  
-co2\_ppmv  
-rtm\_res  
-rtm\_tstep

New precedence for build-namelist options is...

Values set on the command-line using the -namelist option (CLM\_NAMELIST\_OPTS).

Values read from the file specified by -infile (`user_nl_clm` file).  
Datasets from the -clm\_usr\_name option (CLM\_USRDAT\_NAME).  
Values set from a use-case scenario, e.g., -use\_case (CLM\_NML\_USE\_CASE).  
Values from the namelist defaults file.

Namelist options renamed:

`carbon_only` => `suplnitro` (can be set to NONE, PROG\_CROP\_ONLY, or ALL)

Namelist options removed:

`carbon_only` => `suplnitro`  
`scaled_harvest`  
`hist_crtinic`  
`hist_pioflag`  
`ncd_lowmem2d`  
`ncd_pio_def`  
`ncd_pio_UseRearranger`  
`ncd_pio_UseBoxRearr`  
`ncd_pio_SerialCDF`  
`ncd_pio_IODOF_rootonly`  
`ncd_pio_DebugLevel`  
`ncd_pio_num_iotasks`

New history fields:

A5TMIN 5-day running mean of min 2-m temperature (K)  
A10TMIN 10-day running mean of min 2-m temperature (K)  
GDD0 Growing degree days base 0C from planting (ddays)  
GDD8 Growing degree days base 8C from planting (ddays)  
GDD10 Growing degree days base 10C from planting (ddays)  
GDD020 Twenty year average of growing degree days base 0C from planting (ddays)  
GDD820 Twenty year average of growing degree days base 8C from planting (ddays)  
GDD1020 Twenty year average of growing degree days base 10C from planting (ddays)  
GDDPLANT Accumulated growing degree days past planting date for crop (ddays)  
GDDHARV Growing degree days (gdd) needed to harvest (ddays)  
GDDTSOI Growing degree-days from planting (top two soil layers) (ddays)  
QIRRIG water added through irrigation (mm/s)

`SNOWLIQ` and `SNOWICE` changed from average to instantaneous output.

## **What was new with CLM4 in CESM1.0.2 (in CESM1.0.2) since the September 17th, 2010 CESM1.0.1 release?**

Since, CLM4 in CESM1.0.1 in the CESM1.0.1 release there were several developments made to CLM4 in CESM1.0.2. Several new namelist items were added a few new history fields. There were also some updates for running the model with single-point mode.

Configuration options that were renamed:

`prog_seasalt` => `progsslt`

Namelist items removed:

prog\_seasalt => progsst

## **What was new with CLM4 in CESM1.0.2 Science since CLM4 in CESM1.0.1?**

A long simulation at the coarse resolution of T31 (typically used for Paleo-climate studies) was done and an spun-up initial condition file was provided for this resolution (also by default the namelist variable ice\_runoff was turned off for T31). Also a new surface dataset and transient land-cover dataset was provided for half-degree resolution.

## **What was new with CLM4 in CESM1.0.2 Software since CLM4 in CESM1.0.1?**

New configuration options

sitespf\_pt

sitespf\_pt is used for single-point/regional mode and is set to the site-name that will be used (see the config\_definition.xml for the list of valid options).

Configuration options that were renamed:

prog\_seasalt => progsst

Namelist items removed:

faerdep  
fndepdat  
fndepdyn  
use\_ndepstream

Nitrogen deposition datasets are now only entered through the ndepdyn\_nml namelist (removing fndepdat, fndepdyn, and use\_ndepstream). Aerosol deposition is now a required input from the atmosphere model, hence faerdep is removed.

New history fields:

U10 10-m wind (m/s)  
U10\_DUST 10-m wind for dust model (m/s)  
VA atmospheric wind speed plus convective velocity (m/s)  
VOLR RTM storage: LIQ (m<sup>3</sup>)  
VOLR\_ICE RTM storage: ICE (m<sup>3</sup>)

## **What was new with CLM4 in CESM1.0.1 (in CESM1.0.1)**

## **since the April 1st, 2010 CCSM4.0 release?**

From, CLM4.0.00 in the CCSM4.0 release to CLM4 in CESM1.0.1 there were several developments made to CLM. A glacier multiple elevation class option was added that allows the use of CLM4 with a glacier land ice model the Community Ice Sheet Model (CISM). A bug-fix for the snow hydrology was added. Several new namelist items were added a few new history fields. Also the capability of reading aerosol and nitrogen deposition from stream files at one resolution and regridded on the fly rather than with datasets at the model resolution was added in. This was important for higher resolutions so that large datasets do not have to be created before running the model, nor are datasets for every resolution required.

## **What was new with CLM4 in CESM1.0.1 Science since CCSM4.0?**

In general, snow layers should not be thinner than

```
dzmin = wice/rhoice + wliq/rholiq
```

If  $dz < dzmin$ , then the value of "void" computed in subroutine SnowCompaction is negative, which is unphysical. This doesn't cause problems with the compaction itself, but results in unrealistic values of vol\_ice, vol\_liq, and eff\_porosity in subroutine SnowWater. We can have vol\_ice = 1 and vol\_liq = 0 even when liquid is present, which cuts off the runoff (qout) from the lowest snow layer. Liquid water then accumulates in the snow column without draining, which leads to further problems and eventually a code crash.

The solution to this problem was to adjust layer thickness dz for any water+ice content changes in excess of previous layer thickness, e.g.,

```
dz(c, j) = max(dz(c, j), h2osoi_liq(c, j)/denh2o + h2osoi_ice(c, j)/denice)
```

at appropriate steps in the snow hydrology subroutines.

Snow hydrology bug fix.

Add multiple elevation class option for glaciers so can interact with the land ice sheet model.

## **What was new with CLM4 in CESM1.0.1 Software since CCSM4.0?**

New configuration options

glc\_nec

glc\_nec can be 1,3,5, or 10 and MUST match the number on the input surface dataset the elevation classes themselves are read from the surface dataset

New namelist items:

carbon\_only  
create\_glacier\_mec\_landunit  
glc\_dyntopo  
glc\_smb  
ice\_runoff  
ndepmapalgo  
scaled\_harvest

carbon\_only = If true, and CLMCN carbon-nitrogen model is on, Nitrogen is unlimited rather than prognosed and vegetation will be over-productive (replaces the supplemental Nitrogen #ifdef)

create\_glacier\_mec\_landunit (= T when these landunits are created; F by default)

glc\_smb (= T if passing surface mass balance to GLC; else pass PDD info; T by default)

glc\_dyntopo (= T if CLM topography changes dynamically; currently F) (NOT fully implemented yet)

ice\_runoff = If true, river runoff will be split up into liquid and ice streams, otherwise ice runoff will be zero and all runoff directed to liquid stream

ndepmapalgo = Mapping method from Nitrogen deposition input file to the model resolution (can be bilinear,nn,nnoni,nnonj,spval,copy, bilinear by default)

scaled\_harvest = If true, harvesting will be scaled according to coefficients determined by Johann Feddema, 2009

New history fields:

aais\_area Antarctic ice area (km<sup>2</sup>)  
aais\_mask Antarctic mask (unitless)  
gris\_area Greenland ice area (km<sup>2</sup>)  
gris\_mask Greenland mask (unitless)  
QICE ice growth/melt (mm/s)  
QICEYR ice growth/melt (mm/s)  
QTOPSOIL water input to surface (mm/s)  
VOLR RTM storage: LIQ (m<sup>3</sup>)  
VOLR\_ICE RTM storage: ICE (m<sup>3</sup>)

## **What was new with CLM4.0.00 since CLM3.5?**

From CLM3.5 to CLM4.0.00 there were advances in both the science and the software infrastructure. There were also new configure and namelist options as well as new history fields. In this section we will describe each of these changes in turn.

### **What was new with CLM4.0.00 Science?**

The following aspects are changes to the science in CLM4.0.00 since CLM3.5.

#### **Biogeophysics and Hydrology**

Changes to CLM4.0.00 beyond CLM3.5 (Oleson et al., 2008a; Stockli et al., 2008) include updates

throughout the model. The hydrology scheme has been modified with a revised numerical solution of the Richards equation (Zeng and Decker, 2009; Decker and Zeng, 2009); a revised soil evaporation parameterization that removes the soil resistance term introduced in CLM3.5 and replaces it with a so-called *B* formulation, as well as accounts for the role of litter and within-canopy stability (Sakaguchi and Zeng, 2009). CLM4 also includes a representation of the thermal and hydraulic properties of organic soil that operates in conjunction with the mineral soil properties (Lawrence and Slater, 2008). The ground column has been extended to ~50-m depth by adding five additional hydrologically inactive ground layers (making a total of 15 ground layers, 10 soil layers and 5 bedrock layers; Lawrence et al., 2008). An urban landunit and associated urban canyon model (CLMU) has been added which permits the study of urban climate and urban heat island effects (Oleson et al., 2008b).

## **Snow Model**

The snow model is significantly modified via incorporation of SNICAR (SNow and Ice Aerosol Radiation) which represents the effect of aerosol deposition (e.g. black and organic carbon and dust) on albedo, introduces a grain-size dependent snow aging parameterization, and permits vertically resolved snowpack heating (Flanner and Zender, 2005; Flanner and Zender, 2006; Flanner et al., 2007). The new snow model also includes a new density-dependent snow cover fraction parameterization (Niu and Yang, 2007), a revised snow burial fraction over short vegetation (Wang and Zeng, 2009) and corrections to snow compaction (Lawrence and Slater, 2009).

## **Surface Datasets**

The PFT distribution is as in Lawrence and Chase (2007) except that a new cropping dataset is used (Ramankutty et al., 2008) and a grass PFT restriction has been put in place to reduce a high grass PFT bias in forested regions by replacing the herbaceous fraction with low trees rather than grass. Grass and crop PFT optical properties have been adjusted according to values presented in Asner et al. (1998), resulting in significantly reduced albedo biases. Soil colors have been re-derived according to the new PFT distribution.

## **Biogeochemistry**

The model is extended with a carbon-nitrogen biogeochemical model (Thornton et al., 2007; Thornton et al., 2009; Randerson et al., 2009) which is referred to as CLMCN. CN is based on the terrestrial biogeochemistry Biome-BGC model with prognostic carbon and nitrogen cycle (Thornton et al., 2002; Thornton and Rosenbloom, 2005). CLMCN is prognostic with respect to carbon and nitrogen state variables in the vegetation, litter, and soil organic matter. Vegetation phenology and canopy heights are also prognostic. A detailed description of the biogeochemical component can be found in Thornton et al. (2007). Note that CLM4.0.00 can be run with either prescribed satellite phenology (CLMSP) or with prognostic phenology provided by the carbon- nitrogen cycle model (CLMCN). Additionally, a transient land cover and land use change, including wood harvest, capability has been introduced that enables the evaluation of the impact of historic and future land cover and land use change on energy, water, and momentum fluxes as well as carbon and nitrogen fluxes. The dynamic global vegetation model in CLM3.0 has been revised such that the carbon dynamics (e.g. productivity, decomposition, phenology, allocation, etc.) are controlled by CN and only the dynamic vegetation biogeography (competition)

aspect of the CLM3.0 DGVM is retained. The biogenic volatile organic compounds model (BVOC) that was available in CLM3.0 has been replaced with the MEGAN BVOC model (Heald et al. 2008).

## Miscellaneous Changes

Several other minor changes have been incorporated including a change to the atmospheric reference height so that it is the height above  $z_0+d$  for all surface types. The convergence of canopy roughness length  $z_0$  and displacement height  $d$  to bare soil values as the above-ground biomass, or the sum of leaf and stem area indices, goes to zero is ensured (Zeng and Wang, 2007). Several corrections have been made to the way the offline forcing data is interpreted. The main change is a vastly improved and smooth diurnal cycle of incoming solar radiation that conserves the total incoming solar radiation from the forcing dataset. Additionally, in offline mode rather than partitioning incoming solar radiation into a constant 70%/30% direct vs diffuse split, it is partitioned according to empirical equations that are a function of total solar radiation. Finally, to improve global energy conservation in fully coupled simulations, runoff is split into separate liquid and ice water streams that are passed separately to the ocean. Input to the ice water comes from excess snowfall in snow-capped regions.

## Summary of Science Changes

Taken together, these augmentations to CLM3.5 in CLM4.0.00 result in improved soil moisture dynamics that lead to higher soil moisture variability and drier soils. Excessively wet and unvarying soil moisture was recognized as a deficiency in CLM3.5 (Oleson et al. 2008a, Decker and Zeng, 2009). The revised model also simulates, on average, higher snow cover, cooler soil temperatures in organic-rich soils, greater global river discharge, lower albedos over forests and grasslands, and higher transition-season albedos in snow covered regions, all of which are improvements compared to CLM3.5.

## What is new with CLM4.0.00 Software Infrastructure?

The following aspects are changes to the software infrastructure in CLM4.0.00 since CLM3.5.

- Update to cpl7 and scripts.
- Remove offline and cpl6 modes.
- Remove support for CASA model.
- Update to datm8 atmospheric data model.
- Add gx3v7 land mask for T31 and fv-4x5 horizontal resolutions.
- Add gx1v6 land mask for f05, f09, and f19 horizontal resolutions.
- Add tx1v1 land mask and 1.9x2.5\_tx1v1 horizontal resolution.
- Add in 2.5x3.33 horizontal resolution.
- Add in T62 horizontal resolution so can run at same resolution as input DATM data.
- Allow first history tape to be 1D.
- Add ability to use own version of input datasets with CLM\_USRDAT\_NAME variable.
- Add a script to extract out regional datasets.
- New **build-namelist** system with XML file describing all namelist items.
- Add glacier\_mec use-case and stub glacier model.
- Make default of maxpatch\_pft=numpft+1 instead of 4.
- Only output static 3D fields on first h0 history file to save space.
- Add new fields for VOC (Volatile Organic Compounds) on surface datasets, needed for the new MEGAN VOC model.
- Add irrigation area to mksurfdata tool (NOT used in CLM yet).

Add multiple elevation class option for glaciers in mksurfdta tool (NOT used in CLM yet).  
Add ascale field to land model in support of model running on it's own grid.

## What are The New Configuration Options in CLM4.0.00?

Describe any changes made to build system:

Change directory structure to match CCSM.  
Add BGP target.  
Add choice between ESMF and MCT frameworks.  
Start removing #ifdef and directives that supported Cray-X1 Phoenix as now decommissioned.  
Make default of maxpatch\_pft=numpft+1 instead of 4 for all configurations.  
By default turn on CLAMP when either CN or CASA is enabled  
New SNICAR\_FRC, CARBON\_AERO, and C13 CPP ifdef tokens.

New options added to **configure**: More information on options to CLM **configure** are given in the Section called *More information on the CLM configure script* in Chapter 1.

**Option:** -comp\_inf <name>

**Description:** Component interface to use (ESMF or MCT) (default MCT)

**Option:** -nofire

**Description:** Turn off wildfires for bgc setting of CN (default includes fire for CN)

**Option:** -pio <name>

**Description:** Switch enables building with Parallel I/O library. [on | off] (default is on)

**Option:** -snicar\_frc <name>

**Description:** Turn on SNICAR radiative forcing calculation. [on | off] (default is off)

More information on options to CLM **configure** are given in the Section called *More information on the CLM configure script* in Chapter 1.

## What are The New Namelist Options in CLM4.0.00?

**build-namelist** now checks the validity of your namelist you generate by looking at data in the namelist\_definition.xml file. In order to add new namelist items you need to change the code and also edit this file (e.g. a namelist option required for your research project that is not currently an option in CLM4.0.00). To view information on the namelist view the file:

models/lnd/clm/bld/namelist\_files/namelist\_definition.xml in a browser and you'll see the names, type, description and valid\_values for all namelist variables.

Changes to **build-namelist**:

Allow simulation year entered to include ranges of years (i.e. 1850-2000)

Remove cam\_hist\_case option.

Make sure options ONLY used for stand-alone testing have a "drv\_" or "datm\_" prefix in them and list these options all together.

New option to **build-namelist**:

```
-clm_usr_name "name" Dataset resolution/descriptor for personal datasets.  
Default: not used  
Example: 1x1pt_boulderCO_c090722 to describe location,
```

number of pts, and date files created

### New list options to **build-namelist**

```
cd models/lnd/clm/bld
./build-namelist -res list      # List valid resolutions
./build-namelist -mask list     # List valid land-masks
./build-namelist -sim_year list  # List valid simulation years and simulation year ranges
./build-namelist -clm_demand list # List namelist variables including those you could
                                # demand to be set.
./build-namelist -use_case list   # List valid use-cases
```

### New use-cases for **build-namelist**:

```
1850_control = Conditions to simulate 1850 land-use
2000_control = Conditions to simulate 2000 land-use
20thC_transient = Simulate transient land-use, aerosol and Nitrogen deposition
                  from 1850 to 2005
```

### New namelist items:

```
urban_hac = OFF, ON or ON_WASTEHEAT  (default OFF)      Flag for urban Heating
                                         and Air-Conditioning
OFF          = Building internal temperature is un-regulated.
ON           = Building internal temperature is bounded to reasonable range.
ON_WASTEHEAT = Building internal temperature is bounded and resultant waste
                heat is given off.
urban_traffic = .true. or .false.    Flag to include additional multiplicative
                                         factor of urban traffic to sensible heat flux.
                                         (default .false.)
fsnowoptics  = filename        file for snow/aerosol optical properties (required)
fsnowaging   = filename        file for snow aging parameters (required)
```

More information on the **build-namelist** options are given in the Section called *Definition of Namelist items and their default values* in Chapter 1. and in CLM\_BLDNML\_OPTS.

More information on the **build-namelist** options are given in in the Section called *Definition of Namelist items and their default values* in Chapter 1.

## What are The New History Fields?

New history variables: (note watt vs. W in units, 26 vs. 76)

**Name:** BCDEP

**Long-name:** total BC deposition (dry+wet) from atmosphere

**Units:** kg/m<sup>2</sup>/s

**Name:** BIOGENCO

**Long-name:** biogenic CO flux

**Units:** uGC/M2/H

**Name:** C13\_PRODUCT\_CLOSS

**Long-name:** C13 total carbon loss from wood product pools

**Units:** gC13/m<sup>2</sup>/s

**Name:** DSTDEP

**Long-name:** total dust deposition (dry+wet) from atmosphere

**Units:** kg/m<sup>2</sup>/s

**Name:** EFLX\_DYNBAL

**Long-name:** dynamic land cover change conversion energy flux

**Units:** W/m<sup>2</sup>

**Name:** FGR12

**Long-name:** heat flux between soil layers 1 and 2

**Units:** watt/m<sup>2</sup>

**Name:** FSAT

**Long-name:** fractional area with water table at surface

**Units:** unitless

**Name:** FSH\_NODYNLNDUSE

**Long-name:** sensible heat flux not including correction for land use change

**Units:** watt/m<sup>2</sup>

**Name:** GC\_HEAT1

**Long-name:** initial gridcell total heat content

**Units:** J/m<sup>2</sup>

**Name:** GC\_HEAT2

**Long-name:** post land cover change total heat content

**Units:** J/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** GC\_ICE1

**Long-name:** initial gridcell total ice content

**Units:** mm/s

**Name:** GC\_ICE2

**Long-name:** post land cover change total ice content

**Units:** mm/s

**Active/Inactive:** inactive

**Name:** GC\_LIQ1

**Long-name:** initial gridcell total liq content

**Units:** mm

**Name:** GC\_LIQ2

**Long-name:** initial gridcell total liq content

**Units:** mm

**Active/Inactive:** inactive

**Name:** H2OSNO\_TOP

**Long-name:** mass of snow in top snow layer

**Units:** kg

**Name:** HEAT\_FROM\_AC

**Long-name:** sensible heat flux put into canyon due to heat removed from air conditioning

**Units:** watt/m<sup>2</sup>

**Name:** HK

**Long-name:** hydraulic conductivity

**Units:** mm/s

**Active/Inactive:** inactive

**Name:** ISOPRENE

**Long-name:** isoprene flux

**Units:** uGC/M2/H

**Name:** LAND\_USE\_FLUX

**Long-name:** total C emitted from land cover conversion and wood product pools

**Units:** gC/m<sup>2</sup>/s

**Name:** LAND\_UPTAKE

**Long-name:** NEE minus LAND\_USE\_FLUX, negative for update

**Units:** gC/m<sup>2</sup>/s

**Name:** LWup

**Long-name:** upwelling longwave radiation

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** MONOTERP

**Long-name:** monoterpenene flux

**Units:** uGC/M2/H

**Name:** NBP

**Long-name:** net biome production, includes fire, landuse, and harvest flux, positive for sink

**Units:** gC/m<sup>2</sup>/s

**Name:** OCDEP

**Long-name:** total OC deposition (dry+wet) from atmosphere

**Units:** kg/m<sup>2</sup>/s

**Name:** OVOC

**Long-name:** other VOC flux

**Units:** uGC/M2/H

**Name:** ORVOC

**Long-name:** other reactive VOC flux

**Units:** uGC/M2/H

**Name:** PBOT

**Long-name:** atmospheric pressure

**Units:** Pa

**Name:** PCO2

**Long-name:** atmospheric partial pressure of CO<sub>2</sub>

**Units:** Pa

**Name:** PRODUCT\_CLOSS

**Long-name:** total carbon loss from wood product pools

**Units:** gC/m<sup>2</sup>/s

**Name:** PRODUCT\_NLOSS

**Long-name:** total N loss from wood product pools

**Units:** gN/m<sup>2</sup>/s

**Name:** Qair

**Long-name:** atmospheric specific humidity

**Units:** kg/kg

**Active/Inactive:** inactive

**Name:** Qanth

**Long-name:** anthropogenic heat flux

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** Qtau

**Long-name:** momentum flux

**Units:** kg/m/s<sup>2</sup>

**Name:** QFLX\_LIQ\_DYNBAL

**Long-name:** liq dynamic land cover change conversion runoff flux

**Units:** mm/s

**Name:** QFLX\_ICE\_DYNBAL

**Long-name:** ice dynamic land cover change conversion runoff flux

**Units:** mm/s

**Name:** QRUNOFF\_NODYNLNDUSE

**Long-name:** total liquid runoff not including correction for land use change (does not include QSNWCPICE)

**Units:** mm/s

**Name:** QSNWCPICE

**Long-name:** excess snowfall due to snow capping

**Units:** mm/s

**Name:** QSNWCPICE\_NODYNLNDUSE

**Long-name:** excess snowfall due to snow capping not including correction for land use change

**Units:** mm/s

**Name:** QSNWCPLIQ

**Long-name:** excess rainfall due to snow capping

**Units:** mm/s

**Active/Inactive:** inactive

**Name:** SMP

**Long-name:** soil matric potential

**Units:** mm

**Active/Inactive:** inactive

**Name:** SNOAERFRC2L

**Long-name:** surface forcing of all aerosols in snow, averaged only when snow is present (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNOAERFRCL

**Long-name:** surface forcing of all aerosols in snow (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNOBCFRCL

**Long-name:** surface forcing of BC in snow (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNOBCMCL

**Long-name:** mass of BC in snow column

**Units:** kg/m<sup>2</sup>

**Name:** SNOBCMSL

**Long-name:** mass of BC in top snow layer

**Units:** kg/m<sup>2</sup>

**Name:** SNODTdZL

**Long-name:** top snow layer temperature gradient (land)

**Units:** K/m

**Name:** SNODSTFRC2L

**Long-name:** surface forcing of dust in snow, averaged only when snow is present (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNODSTFRCL

**Long-name:** surface forcing of dust in snow (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNODSTMCL

**Long-name:** mass of dust in snow column

**Units:** kg/m<sup>2</sup>

**Name:** SNODSTMSL

**Long-name:** mass of dust in top snow layer

**Units:** kg/m<sup>2</sup>

**Name:** SNOFSRND

**Long-name:** direct nir reflected solar radiation from snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSRNI

**Long-name:** diffuse nir reflected solar radiation from snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSRVD

**Long-name:** direct vis reflected solar radiation from snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSRVI

**Long-name:** diffuse vis reflected solar radiation from snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSDSND

**Long-name:** direct nir incident solar radiation on snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSDSNI

**Long-name:** diffuse nir incident solar radiation on snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSDSVD

**Long-name:** direct vis incident solar radiation on snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOFSDSVI

**Long-name:** diffuse vis incident solar radiation on snow

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** SNOLIQFL

**Long-name:** top snow layer liquid water fraction (land)

**Units:** fraction

**Active/Inactive:** inactive

**Name:** SNOOCMCL

**Long-name:** mass of OC in snow column

**Units:** kg/m<sup>2</sup>

**Name:** SNOOCMSL

**Long-name:** mass of OC in top snow layer

**Units:** Kg/m<sup>2</sup>

**Name:** SNOOCFRC2L

**Long-name:** surface forcing of OC in snow, averaged only when snow is present (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNOOCFRCL

**Long-name:** surface forcing of OC in snow (land)

**Units:** watt/m<sup>2</sup>

**Name:** SNORDSL

**Long-name:** top snow layer effective grain radius

**Units:** m<sup>-6</sup>

**Active/Inactive:** inactive

**Name:** SNOTTOPL

**Long-name:** snow temperature (top layer)

**Units:** K/m

**Active/Inactive:** inactive

**Name:** SWup

**Long-name:** upwelling shortwave radiation

**Units:** watt/m<sup>2</sup>

**Active/Inactive:** inactive

**Name:** TSOI\_10CM

**Long-name:** soil temperature in top 10cm of soil

**Units:** K

**Name:** URBAN\_AC

**Long-name:** urban air conditioning flux

**Units:** watt/m<sup>2</sup>

**Name:** URBAN\_HEAT

**Long-name:** urban heating flux

**Units:** watt/m<sup>2</sup>

**Name:** VOCFLXT

**Long-name:** total VOC flux into atmosphere

**Units:** uGC/M2/H

**Name:** Wind

**Long-name:** atmospheric wind velocity magnitude

**Units:** m/s

**Active/Inactive:** inactive

**Name:** WOOD\_HARVESTC

**Long-name:** wood harvest (to product pools)

**Units:** gC/m<sup>2</sup>/s

**Name:** WOOD\_HARVEST

**Long-name:** wood harvest (to product pools)

**Units:** gN/m<sup>2</sup>/s

History field name changes:

**Old:** ANNSUM\_PLANT\_NDEMAND

**New:** = ANNSUM\_POTENTIAL\_GPP

**Old:** ANNSUM\_RETRANSN

**New:** = ANNMAX\_RETRANSN

**Old:** C13\_DWT\_PROD10C\_LOSS

**New:** = C13\_PROD10C\_LOSS

**Old:** C13\_DWT\_PROD100C\_LOSS

**New:** = C13\_PROD100C\_LOSS

**Old:** C13\_DWT\_PROD10N\_LOSS

**New:** = C13\_PROD10N\_LOSS

**Old:** C13\_DWT\_PROD100C\_LOSS

**New:** = C13\_PROD100C\_LOSS

**Old:** DWT\_PROD100N\_LOSS

**New:** = PROD10N\_LOSS

**Old:** DWT\_PROD100N\_LOSS

**New:** = PROD100N\_LOSS

**Old:** DWT\_PROD100C\_LOSS

**New:** = PROD10C\_LOSS

**Old:** DWT\_PROD100C\_LOSS

**New:** = PROD100C\_LOSS

**Old:** HCSOISNO

**New:** = HC

**Old:** TEMPSUM\_PLANT\_NDEMAND

**New:** = TEMPSUM\_POTENTIAL\_GPP

**Old:** TEMPSUM\_RETRANSN

**New:** = TEMPMAX\_RETRANSN

History field names deleted include: SNOWAGE, TSNOW, FMICR, FCO2, DMI, QFLX\_SNOWCAP

Add new urban oriented \_U, and \_R (Urban and Rural) for the following history variables:

EFLX\_LH\_TOT, FGR, FIRA, FSH, FSM, Q2M, QRUNOFF, RH2M, SoilAlpha, TG, TREFMNAV, TREFMXAV, and TSA (missing \_R for SoilAlpha as the regular SoilAlpha is only defined for rural areas anyway)

**Note:** We are missing the Rural soil-alpha variable: SoilAlpha\_R on purpose. SoilAlpha\_U is only defined over pervious road, and missing everywhere else. SoilAlpha is defined only for rural areas.

# Quickstart to using CLM4

Before working with CLM4 read the QuickStart Guide in the CESM1.0.4 Scripts User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>). Once you are familiar with how to setup cases for any type of simulation with CESM you will want to direct your attention to the specifics of using CLM.

For some of the details of setting up cases for CLM4 read the README and text files available from the "models/lnd/clm/doc" directory (see the "CLM Web pages" section for a link to the list of these files). Here are the important ones that you should be familiar with.

1. README (./README) file describing the directory structure.
2. Quickstart.userdatasets (./Quickstart.userdatasets) file describing how to use your own datasets in the model (also see the Section called *Creating your own single-point/regional surface datasets* in Chapter 5).
3. models/lnd/clm/doc/KnownBugs (./KnownBugs) file describing known problems in CLM4 (that we expect to eventually fix).
4. models/lnd/clm/doc/KnownLimitations (./KnownLimitations) file describing known limitations in CLM4 and workarounds that we do NOT expect to fix.

The *IMPORTANT\_NOTES* file talks about important things for users to know about using the model scientifically. Its content is given in the next chapter on "*What is scientifically validated and functional in CLM4?*".

The *ChangeLog/ChangeSum* talk about advances in different versions of CLM. The content of these files is largely explained in the previous chapter on "*What is new with CLM4 in CESM1.0.4 since previous public releases?*".

Note other directories have README files that explain different components and tools used when running CLM and are useful in understanding how those parts of the model work and should be consulted when using tools in those directories. For more details on configuring and customizing a case with CLM see Chapter 1.

The *Quickstart.GUIDE* (which can be found in `models/lnd/clm/doc`) is repeated here.

```
Quick-Start to Using cpl7 Scripts for clm4
=====
Assumptions: You want to use bluefire with clm4
              to do a clm simulation with data atmosphere and the
              latest atm forcing files and settings. You also want to cycle
              the atm data between 1948 to 2004 and you want to run at
              1.9x2.5 degree resolution.

Process:

# Create the case

cd scripts

./create_newcase -case <testcase> -mach bluefire -res f19_g16 -compset i4804
(./create_newcase -help -- to get help on the script)
TRUE

# Configure the case

cd <testcase>
$EDITOR env_run.xml env_conf.xml    # If you need to make changes (or use the xmlchange script)
```

```

./configure -case
(./configure -help -- to get help on the script)

# Make any changes to the namelist

$EDITOR Buildconf/clm.buildnml_prestage.csh

# Compile the code

./<testcase>.bluefire.build

# Submit the run

./<testcase>.bluefire.submit

Information on Compsets:

"I" compsets are the ones with clm and datm7 without ice and ocean.
The latest "I" compsets use the new CLM_QIAN data with solar following
the cosine of solar zenith angle, precipitation constant, and other
variables linear interpolated in time (and with appropriate time-stamps on
the date). Some of the I compsets are:

Name      (short-name): Description
-----
I_2000      (I):          CLM to simulate year=2000
I_1850      (I1850):       CLM to simulate year=1850
I_1948_2004  (I4804):     CLM running with atm data over 1948-2004
I_1850-2000  (I8520):     CLM with transient PFT over 1850-2000
I_2000_CN    (ICN):        CLM with CN on to simulate year=2000
I_1850_CN    (I1850CN):   CLM with CN on to simulate year=1850
I_1948-2004_CN (I4804CN): CLM with CN on running with atm data over 1948-2004
I_1850-2000_CN (I8520CN): CLM with CN on with transient PFT over 1850-2000

Automatically resubmitting jobs:

After doing a short simulation that you believe is correct

./xmlchange -file env_run.xml -id CONTINUE_RUN -val TRUE

# Change RESUBMIT to number greater than 0, and CONTINUE_RUN to TRUE...

./<testcase>.bluefire.submit

```

# **What is scientifically validated and functional in CLM4?**

In this section we go over what has been extensively tested and scientifically validated with CLM4, and maybe more importantly what has NOT been tested and may NOT be scientifically validated. You can use all features of CLM, but need to realize that some things haven't been tested extensively or validated scientifically. When you use these features you may run into trouble doing so, and will need to do your own work to make sure the science is reasonable.

## **Standard Configuration and Namelist Options that are Validated**

The standard version of the model is CLMCN at 1-degree horizontal resolution (0.9x1.25). This version has been scientifically validated with long simulations for: fully coupled simulations ("B" cases), coupled to atmosphere model CAM ("F" cases), and stand-alone CLM cases ("I" cases). We've also done both long simulations for 1850 conditions, and transient 20th century simulations from 1850 to 2005 (with transient land-use, Nitrogen and Aerosol deposition). There have also been transient future scenario simulations done for fully coupled cases for different "representative concentration pathway" (RCP) scenarios (RCP2.6, RCP4.5, RCP6.0, and RCP8.5). To a lesser extent there have also been simulations done at T31 and 2-degree horizontal resolution (1.9x2.5), and with CLMSP for these resolutions. As such we have provided appropriate 1-degree, 2-degree, and T31 initial condition datasets for these configurations. The irrigation and prognostic crop models were both validated at 2-degree resolution. The irrigation model for CLMSP for present day conditions for an "I" compset, and the prognostic crop model for present day conditions for a case coupled to the active land model, but using a data ocean model (an "F" compset). Other resolutions, configurations, and namelist options are less well tested or scientifically validated. The further you get away from the standard configurations and resolutions, the more likely you are to run into trouble, and/or need to scientifically validate your work.

In the sections below we go through configuration and/or namelist options or modes that the user should be especially wary of using. You are of course free to use these options, and you may find that they work functionally. Although in some cases you will find issues even with functionality of using them. If so you will need to test, debug and find solutions for these issues on your own. But in every case you will need to go through more extensive work to validate these options from a scientific standpoint.

## **Configure Modes NOT scientifically validated, documented, supported or, in some cases, even advised**

## **to be used:**

### 1. C13

(-c13)

The C13 mode for bgc=cn is NOT scientifically validated or documented and is NOT recommended for use.

### 2. CASA

(-bgc casa)

The bgc=casa mode is NOT scientifically validated or documented and is NOT recommended for use.

### 3. SNICAR\_FRC

(-snicar\_frc)

This mode is tested and functional, but is NOT constantly scientifically validated, and should be considered experimental.

## **Namelist options that should NOT be exercised:**

### **Build-Namelist options that should NOT be exercised:**

#### 1. *-irrig with -bgc cn*

We have only run the irrigation model with CLMSP (i.e. without the CN model). We recommend that if you want to run the irrigation model with CN, that you do a spinup. But, more than that you may need to make adjustments to `irrig_factor` in `models/lnd/clm/src/biogeophys/CanopyFluxesMod.F90`. See the notes on this in the description of the irrigation model in the Technical Descriptions of the Interactive Crop Management and Interactive Irrigation Models (<http://www.cesm.ucar.edu/models/cesm1.0//CLMcropANDirrigTechDescriptions.pdf>).

#### 2. *-irrig with -crop on*

Irrigation doesn't work with the prognostic crop model. Irrigation is only applied to generic crop currently, which negates it's practical usage. We also have a known problem when both are on (see bug 1326 in the `models/lnd/clm/doc/KnownBugs` (`./KnownBugs`) file). If you try to run in this mode, the CLM **build-namelist** will return with an error.

#### 3. *-lnd\_res*:

Fine-mesh mode, functional, but experimental

#### 4. *-rcp*:

Representative Concentration Pathway (RCP) for future scenarios, functional for limited resolutions, but experimental

#### 5. *-datm\_\**:

All options that start with "datm\_" they are only used for CLM stand-alone testing.

#### 6. *-drv\_\**:

All options that start with "drv\_" they are only used for CLM stand-alone testing.

## **Namelist items that should NOT be exercised:**

1. *casa namelist options*: lnpp, lalloc, q10, spunup, and fcpool CASA has NOT been scientifically validated in CLM4.
2. *fine-mesh namelist options*: flndtopo, and fatmtopo. These options are functional but experimental. See the -lnd\_res option above.
3. *suplnitro='ALL'* The suplnitro namelist option to the CN Biogeochemistry model supplies unlimited nitrogen and therefore vegetation is over-productive in this mode.
4. *urban\_traffic*: Not currently functional

# What are the UNIX utilities required to use CLM?

Running the CLM requires a suite of UNIX utilities and programs and you should make sure you have all of these available before trying to go forward with using it. If you are missing one of these you should contact the systems administrator for the machine you wish to run on and make sure they are installed.

FORTRAN-90 compiler  
"C" compiler  
GNU make  
UNIX csh and tcsh shells  
UNIX sh shell  
UNIX bash shell  
UNIX awk  
UNIX sed  
NetCDF library  
MPI Library  
"C" pre-processor  
Perl  
Autoconf  
m4 macro processor  
Parallel NetCDF (optional)  
NCL (for some of the offline tools for creating/modifying CLM input datasets see Chapter 2 for more information on NCL)  
Python (optional, needed for PTCLM)  
xsltproc, docbook and docbook utilities (optional, needed to build the Users-Guide)  
protex and latex2html (optional, needed to build the Code-Reference Guide)

# Other resources to get help from

In addition to this users-guide there are several other resources that are available to help you use CLM4. The first one is the CESM User's-Guide, which documents the entire process of creating cases with CESM. The next is the CESM bulletin board which is a web-site for exchanging information between users of CESM. There are also CLM web-pages specific for CLM, and finally there is an email address to report bugs that you find in CESM1.0.

## The CESM User's-Guide

CLM4 in CESM1.0.4 is always run from within the standard CESM1.0.4 build and run scripts. Therefore, the user of CLM4 should familiarize themselves with the CESM1.0.4 scripts and understand how to work with them. User's-Guide documentation on the CESM1.0.4 scripts are available from the following web-page. The purpose of this CLM4 in CESM1.0.4 User's Guide is to give the CLM4 user more complete details on how to work with CLM and the set of tools that support CLM, as well as to give examples that are unique to the use of CLM. However, the CESM1.0.4 Scripts User's-Guide remains the primary source to get detailed information on how to build and run the CESM system.

CESM1.0 Scripts User's-Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>)

## The CESM Bulletin Board

There is a rich and diverse set of people that use the CESM, and often it is useful to be in contact with others to get help in solving problems or trying something new. To facilitate this we have an online Bulletin Board for questions on the CESM. There are also different sections in the Bulletin Board for the different component models or for different topics.

CESM Online Bulletin Board (<http://bb.cgd.ucar.edu/>)

## The CLM web pages

The main CLM web page contains information on the CLM, it's history, developers, as well as downloads for previous model versions. There are also documentation text files in the models/lnd/clm/doc directory that give some quick information on using CLM.

CLM web page (<http://www.cgd.ucar.edu/tss/clm/>)  
CLM Documentation Text Files (..)

Also note that several of the XML database files can be viewed in a web browser to get a nice table of namelist options, namelist defaults, or compsets. Simply view them as a local file and bring up one of the following files:

models/lnd/clm/bld/namelist\_files/namelist\_definition.xml (../../bld/namelist\_files/namelist\_definition.xml) -- definition of  
models/lnd/clm/bld/namelist\_files/namelist\_defaults\_clm.xml (../../bld/namelist\_files/namelist\_defaults\_clm.xml) -- defau

scripts/ccsm\_utils/Case.template/config\_definition.xml (../../../../../scripts/ccsm\_utils/Case.template/config\_definition.xml)  
scripts/ccsm\_utils/Case.template/config\_compsets.xml (../../../../../scripts/ccsm\_utils/Case.template/config\_compsets.xml)   
models/lnd/clm/bld/namelist\_files/history\_fields.xml (../../bld/namelist\_files/history\_fields.xml) -- definition of CLM histo

## Reporting bugs in CLM4

If you have any problems, additional questions, bug reports, or any other feedback, please send an email to <cesmhelp@cgd.ucar.edu>. If you find bad, wrong, or misleading information in this users guide send an email to <erik@ucar.edu>. The current list of known issues for CLM4 in CESM1.0.4 is in the models/lnd/clm/doc/KnownBugs (./KnownBugs) file, and the list of issues for CESM1.0.4 is at...

[http://www.cesm.ucar.edu/models/cesm1.0//tags/cesm1\\_0\\_4/#PROBLEMS](http://www.cesm.ucar.edu/models/cesm1.0//tags/cesm1_0_4/#PROBLEMS)  
([http://www.cesm.ucar.edu/models/cesm1.0//tags/cesm1\\_0\\_4/#PROBLEMS](http://www.cesm.ucar.edu/models/cesm1.0//tags/cesm1_0_4/#PROBLEMS)).

# Chapter 1. How to customize the configuration for a case with CLM

The CESM User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>) gives you the details on how to setup, **configure**, build, and run a case. That is the document to give you the details on using the CESM scripts. The purpose of this document is to give you the details when using CESM with CLM on how to customize and use advanced features in CLM. You should be familiar with the CESM User's Guide and how to setup cases with CESM1.0.4 before referring to this document.

In this chapter we deal with three different ways of customizing a case: Choosing a compset, Customizing Configuration options, and customizing the CLM Namelist. There are many different compsets that use CLM and many are setup to enable special features of CLM from the start. So the first thing you want to be familiar with are the different options in the compsets. The next section shows the different options for customizing the configuration options for CLM. Here we introduce the CLM **configure** and **build-namelist** scripts and how using the options in `env_conf.xml` you can customize the configuration and the initial namelist. The final section tells you about the CLM namelist and how you can customize the namelist once you have run "**configure -case**" and have an initial namelist in `BuildConf/clm.buildnml.csh`. You can also use `env_conf.xml` options to change your namelist as well, before "**configure -case**" is run.

## Choosing a compset using CLM

When setting up a new case one of the first choices to make is which "component set" (or compset) to use. The compset refers to which component models are used as well as specific settings for them. We label the different types of compsets with a different letter of the alphabet from "A" (for all data model) to "X" (for all dead model). The compsets of interest when working with CLM are the "I" compsets (which contain CLM with a data atmosphere model and a stub ocean, and stub sea-ice models), "E" and "F" compsets (which contain CLM with the active atmosphere model (CAM), prescribed sea-ice model, and a data ocean model), and "B" compsets which have all active components. Below we go into details on the "I" compsets which emphasize CLM as the only active model, and just mention the two other categories.

When working with CLM you usually want to start with a relevant "I" compset before moving to the more complex cases that involve other active model components. The "I" compsets can exercise CLM in a way that is similar to the coupled modes, but with much lower computational cost and faster turnaround times.

### Compsets coupled to data atmosphere and stub ocean/sea-ice ("I" compsets)

1. `I_2000(I`) Active land model with QIAN atm input data for 1972-2004 and Satellite phenology (SP), CO<sub>2</sub> level and Aerosol deposition for 2000
2. `I_1850(I1850`) Active land model with QIAN atm input data for 1948 to 1972 and Satellite phenology (SP), CO<sub>2</sub> level and Aerosol deposition for 1850

3. `I_1850_CN(I1850CN)` Active land model with QIAN atm input data for 1948 to 1972 and CN (Carbon Nitrogen) biogeochemistry, CO<sub>2</sub> level and Aerosol deposition for 1850
4. `I_1850_SPINUP_3HrWx_CN(I1850SPINUPCN)` Active land model with BCN CPLHIST 3-hourly weather forcing data and half-hourly solar for 1850 spinup of CN (Carbon Nitrogen) biogeochemistry, CO<sub>2</sub> level and Aerosol deposition for 1850
5. `I_1850-2000(I20TR)` Active land model with QIAN atm input data for 1948 to 1972 and transient Satellite phenology (SP), and Aerosol deposition from 1850 to 2000 and 2000 CO<sub>2</sub> level
6. `I_1850-2000_CN(I20TRCN)` Active land model with QIAN atm input data for 1948 to 1972 and transient CN, Aerosol dep from 1850 to 2000 and 2000 CO<sub>2</sub> level
7. `I_1948-2004(I4804)` Active land model with QIAN atm input data for 1948 to 2004 and Satellite phenology (SP), CO<sub>2</sub> level and Aerosol deposition for 2000
8. `I_1948-2004_CN(I4804CN)` Active land model with QIAN atm input data for 1948 to 2004 and CN (Carbon Nitrogen) biogeochemistry, CO<sub>2</sub> level and Aerosol deposition for 2000
9. `I_2000_CN(ICN)` Active land model with QIAN atm input data for 1972-2004 and CN (Carbon Nitrogen) biogeochemistry, CO<sub>2</sub> level and Aerosol deposition for 2000
10. `I_TEST_2003_CN(ICNTTEST)` Test active land model with QIAN atm input data for just 2002-2003 and CN (Carbon-Nitrogen), CO<sub>2</sub> level and Aerosol deposition for 2000 (BECAUSE OF THE SHORT FORCING PERIOD -- DO NOT USE FOR LONG SIMULATIONS)
11. `I_2000_GLC(IG)` Active glacier model and active land model with QIAN atm input data for 1972-2004 and Satellite phenology (SP), CO<sub>2</sub> level and Aerosol deposition for 2000
12. `I_1850_GLC(IG1850)` Active land and glacier model with QIAN atm input data for 1948 to 1972 and Satellite phenology (SP), CO<sub>2</sub> level and Aerosol deposition for 1850
13. `I_1850_CN_GLC(IG1850CN)` Active land and glacier model with QIAN atm input data for 1948 to 1972 and CN (Carbon Nitrogen) biogeochemistry, CO<sub>2</sub> level and Aerosol deposition for 1850
14. `I_1850-2000_GLC(IG20TR)` Active land and glacier model with QIAN atm input data for 1948 to 1972 and transient Satellite phenology (SP), and Aerosol deposition from 1850 to 2000 and 2000 CO<sub>2</sub> level
15. `I_1850-2000_CN_GLC(IG20TRCN)` Active land and glacier model with QIAN atm input data for 1948 to 1972 and transient CN, Aerosol dep from 1850 to 2000 and 2000 CO<sub>2</sub> level
16. `I_2000_CN_GLC(IGCN)` Active land and glacier model with QIAN atm input data for 1972-2004 and CN (Carbon Nitrogen) biogeochemistry, CO<sub>2</sub> level and Aerosol deposition for 2000
17. `I_RCP2.6_CN_GLC(IGRCP26CN)` Active land and glacier model, RCP2.6 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO<sub>2</sub> level
18. `I_RCP4.5_CN_GLC(IGRCP45CN)` Active land and glacier model, RCP4.5 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO<sub>2</sub> level
19. `I_RCP6.0_CN_GLC(IGRCP60CN)` Active land and glacier model, RCP6.0 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO<sub>2</sub> level
20. `I_RCP8.5_CN_GLC(IGRCP85CN)` Active land and glacier model, RCP8.5 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO<sub>2</sub> level
21. `I_RCP2.6_CN(IRCP26CN)` Active land model, RCP2.6 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO<sub>2</sub> level

22. I\_RCP4.5\_CN(IRCP45CN) Active land model, RCP4.5 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO2 level
23. I\_RCP6.0\_CN(IRCP60CN) Active land model, RCP6.0 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO2 level
24. I\_RCP8.5\_CN(IRCP85CN) Active land model, RCP8.5 future scenario, with CN in CLM, QIAN atm input data for 1972 to 2004, 2000 CO2 level
25. I\_TEST\_2003(ATEST) Test active land model with QIAN atm input data for just 2002-2003 and Satellite phenology (SP), CO2 level and Aerosol deposition for 2000 (BECAUSE OF THE SHORT FORCING PERIOD -- DO NOT USE FOR LONG SIMULATIONS)

## **Compsets coupled to active atmosphere with data ocean**

CAM compsets are compsets that start with "E" or "F" in the name. They are described more fully in the scripts documentation or the CAM documentation. "E" compsets have a slab ocean model while "F" compsets have a data ocean model.

## **Fully coupled compsets with fully active ocean, sea-ice, and atmosphere**

Fully coupled compsets are compsets that start with "B" in the name. They are described more fully in the scripts documentation.

## **Conclusion to choosing a compset**

We've introduced the basic type of compsets that use CLM and given some further details for the "standalone CLM" (or "I" compsets). The `config_compsets.xml` (`../../../../scripts/ccsm_utils/Case.template/config_compsets.xml`) lists all of the compsets and gives a full description of each of them. In the next section we look into customizing the **configure** time options for compsets using CLM.

# **Customizing the CLM configuration**

The "Creating a Case" section of the CESM1.0.4 Scripts User's-Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>) gives instructions on creating a case. What is of interest here is how to customize your use of CLM for the case that you created. In this section we discuss how to customize your case before the first step -- the "**configure** -case" step is done. In the next section we will discuss how to customize your CLM namelist after "**configure** -case" has already been done.

For CLM when "**configure** -case" is called there are two steps that take place:

1. The CLM "**configure**" script is called to setup the build-time configuration for CLM (more information on **configure** is given in the Section called *More information on the CLM **configure** script*).
2. The CLM "**build-namelist**" script is called to generate the initial run-time namelist for CLM (more information on **build-namelist** is given below in the Section called *Definition of Namelist items and their default values*).

When customizing your case at the **configure** step you are able to modify the process by effecting either one or both of these steps. The CLM "**configure**" and "**build-namelist**" scripts are both available in the "models/lnd/clm/bld" directory in the distribution. Both of these scripts have a "-help" option that is useful to examine to see what types of options you can give either of them.

There are five different types of customization for the configuration that we will discuss: CESM1.0 CLM configuration items, Configure time User Namelist, other noteworthy CESM configuration items, the CLM **configure** script options, and the CLM **build-namelist** script options.

Information on all of the script, configuration, build and run items is found under `scripts/ccsm_utils/Case.template` in the `config_definition.xml` (`../../../../scripts/ccsm_utils/Case.template/config_definition.xml`) file.

## CLM Script configuration items

Below we list each of the CESM configuration items that are specific to CLM. All of these are available in your: `env_conf.xml` file.

`CLM_CONFIG_OPTS`  
`CLM_BLDNML_OPTS`  
`CLM_NAMELIST_OPTS`  
`CLM_FORCE_COLDSTART`  
`CLM_NML_USE_CASE`  
`CLM_PT1_NAME`  
`CLM_USRDAT_NAME`  
`CLM_CO2_TYPE`

For the precedence of the different options to **build-namelist** see the section on precedence below.

The first item `CLM_CONFIG_OPTS` has to do with customizing the CLM configuration options for your case, the rest all have to do with generating the initial namelist.

### `CLM_CONFIG_OPTS`

The option `CLM_CONFIG_OPTS` is all about passing command line arguments to the CLM **configure** script. It is important to note that some compsets, may already put a value into the `CLM_CONFIG_OPTS` variable. You can still add more options to your `CLM_CONFIG_OPTS` but make sure you add to what is already there rather than replacing it. Hence, we recommend using the "-append" option to the `xmlchange` script. In the Section called *More information on the CLM **configure** script* below we will go into more details on options that can be customized in the CLM "**configure**" script. It's also important to note that the CLM template may already invoke certain CLM **configure** options and as such those command line options are NOT going to be available to change at this step (nor would you want to change them). The options to **configure** are given with the "-help" option which is given in the Section called *More information on the CLM **configure** script*.

## CLM\_NML\_USE\_CASE

CLM\_NML\_USE\_CASE is used to set a particular set of conditions that set multiple namelist items, all centering around a particular usage of the model. To list the valid options do the following:

```
> cd models/lnd/clm/doc
> ../../build/namelist -use_case list
```

The output of the above command is:

```
build-namelist - use cases: 1850-2100_rcp2.6_glacierMEC_transient 1850-2100_rcp2.6_transient \
1850-2100_rcp4.5_glacierMEC_transient 1850-2100_rcp4.5_transient \
1850-2100_rcp6_glacierMEC_transient 1850-2100_rcp6_transient \
1850-2100_rcp8.5_glacierMEC_transient 1850-2100_rcp8.5_transient 1850_control \
1850_glacierMEC_control 2000-2100_rcp8.5_transient 2000_control 2000_glacierMEC_control \
20thC_glacierMEC_transient 20thC_transient glacierMEC_pd pergro0_pd pergro_pd stdurbpt_pd
Use cases are:...

1850-2100_rcp2.6_glacierMEC_transient = Simulate transient land-use, and aerosol deposition changes \
with historical data from 1850 to 2005 and then with the RCP2.6 scenario from IMAGE

1850-2100_rcp2.6_transient = Simulate transient land-use, and aerosol deposition changes with \
historical data from 1850 to 2005 and then with the RCP2.6 scenario from IMAGE

1850-2100_rcp4.5_glacierMEC_transient = Simulate transient land-use, and aerosol deposition changes \
with historical data from 1850 to 2005 and then with the RCP4.5 scenario from MINICAM

1850-2100_rcp4.5_transient = Simulate transient land-use, and aerosol deposition changes with \
historical data from 1850 to 2005 and then with the RCP4.5 scenario from MINICAM

1850-2100_rcp6_glacierMEC_transient = Simulate transient land-use, and aerosol deposition changes \
with historical data from 1850 to 2005 and then with the RCP6 scenario from AIM

1850-2100_rcp6_transient = Simulate transient land-use, and aerosol deposition changes with \
historical data from 1850 to 2005 and then with the RCP6 scenario from AIM

1850-2100_rcp8.5_glacierMEC_transient = Simulate transient land-use, and aerosol deposition changes \
with historical data from 1850 to 2005 and then with the RCP8.5 scenario from MESSAGE

1850-2100_rcp8.5_transient = Simulate transient land-use, and aerosol deposition changes with \
historical data from 1850 to 2005 and then with the RCP8.5 scenario from MESSAGE

1850_control = Conditions to simulate 1850 land-use
1850_glacierMEC_control = Running an IG case for 1850 conditions with the ice sheet model glimmer
2000-2100_rcp8.5_transient = Simulate transient land-use, and aerosol deposition changes with \
historical data from 2000 to 2005 and then with the RCP8.5 scenario from MESSAGE

2000_control = Conditions to simulate 2000 land-use
2000_glacierMEC_control = Running an IG case for 2000 conditions with the ice sheet model glimmer
20thC_glacierMEC_transient = Simulate transient land-use, and aerosol deposition changes from 1850 \
to 2005
20thC_transient = Simulate transient land-use, and aerosol deposition changes from 1850 to 2005
    glacierMEC_pd = Running an IG case with the ice sheet model glimmer
    pergro0_pd = Perturbation error growth test with unperturbed initial conditions
    pergro_pd = Perturbation error growth test with initial conditions perturbed by roundoff \
level
    stdurbpt_pd = Standard Urban Point Namelist Settings
```

**Note:** See the the Section called *Precedence of Options* section for the precedence of this option relative to the others.

## CLM\_BLDNML\_OPTS

The option CLM\_BLDNML\_OPTS is for passing options to the CLM "build-namelist" script. As with the "configure" script the CLM template may already invoke certain options and as such those options will NOT be available to be set here. The best way to see what options can be sent to the "build-namelist" script is to do

```
> cd models/lnd/clm/bld  
> ./build-namelist -help
```

Here is the output from the above.

```
./
```

The CLM template already sets the resolution and mask as well as the **configure** file, the start-type, the co2\_ppmv, glc\_grid, rtm\_tstep, and rtm\_res, and defines an input namelist and namelist input file, and it normally sets either "-ignore\_ic\_year" or "-ignore\_ic\_date". Also many of the options are designed solely for CLM stand-alone testing and hence should NOT be used (any of the options starting with a "datm\_" or "drv\_" prefix. Hence there are then only five different options that could be set:

1. -irrig
2. -lnd\_res
3. -sim\_year
4. -rcp
5. -clm\_demand
6. -verbose

When "-irrig" is used **build-namelist** will try to find surface datasets that have the irrigation model enabled.

"-lnd\_res" is used to run CLM in fine-mesh mode at a higher resolution than the atmospheric model. This can be useful to get higher resolution from the land model, but saving computer time by running the more expensive atmospheric model at a lower resolution. To get a list of valid resolutions to run at do the following:

```
> cd models/lnd/clm/doc  
> ../../bld/build-namelist -lnd_res list
```

### Caution

The fine-mesh mode is considered experimental, and you may run into problems when you use it. Another option is to use the CESM level "tri-grid" capability to run the land model on a different grid than the atmospheric model. Read the CESM User's-Guide to learn how to do this.

**Note:** See the the Section called *Precedence of Options* section for the precedence of this option relative to the others.

"-clm\_demand" asks the **build-namelist** step to require that the list of variables entered be set. Typically, this is used to require that optional filenames be used and ensure they are set before continuing. For example, you may want to require that `fpftdyn` be set to get dynamically changing vegetation types. To do this you would do the following.

```
> ./xmlchange -file env_conf.xml -id CLM_BLDNML_OPTS -val "-clm_demand fpftdyn"
```

To see a list of valid variables that you could set do this:

```
> cd models/lnd/clm/doc  
> ../../bld/build-namelist -clm_demand list
```

**Note:** Using a 20th-Century transient compset or the 20thC\_transient use-case using `CLM_NML_USE_CASE` would set this as well, but would also use dynamic nitrogen and aerosol deposition files, so using `-clm_demand` would be a way to get *just* dynamic vegetation types and NOT the other files as well.

"-sim\_year" is used to set the simulation year you want the data-sets to simulate conditions for in the input datasets. The simulation "year" can also be a range of years in order to do simulations with changes in the dataset values as the simulation progresses. To list the valid options do the following:

```
> cd models/lnd/clm/doc  
> ../../bld/build-namelist -sim_year list
```

"-rcp" is used to set the representative concentration pathway for the future scenarios you want the data-sets to simulate conditions for, in the input datasets. To list the valid options do the following:

```
> cd models/lnd/clm/doc  
> ../../bld/build-namelist -rcp list
```

## CLM\_NAMELIST\_OPTS

The option `CLM_NAMELIST_OPTS` is for passing namelist items into the "clm\_inparm" namelist. Any items that are set in `CLM_NAMELIST_OPTS` will be set in your namelist after "`configure -case`" is done.

**Important:** For character namelist items you need to use "&apos;" as quotes for strings so that the scripts don't get confused with other quotes they use.

Example, you want to set `hist_dov2xy` to `.false.` so that you get vector output to your history files. To do so edit `env_conf.xml` and add a setting for `hist_dov2xy`. So do the following:

```
> ./xmlchange -file env_conf.xml -id CLM_NAMELIST_OPTS -val hist_dov2xy=.false.
```

Example, you want to set `hist_fincl1` to add the variable '`HK`' to your history files. To do so edit `env_conf.xml` and add a setting for `hist_fincl1`. So do the following:

```
> ./xmlchange -file env_conf.xml -id CLM_NAMELIST_OPTS -val "hist_fincll='';HK'"
```

For a list of the history fields available see CLM History Fields  
`(../../bld/namelist_files/history_fields.xml)`.

**Note:** See the the Section called *Precedence of Options* section for the precedence of this option relative to the others.

### CLM\_CO2\_TYPE

CLM\_CO2\_TYPE sets the type of input CO<sub>2</sub> for either "constant", "diagnostic" or prognostic". If "constant" the value from CCSM\_CO2\_PPMV will be used. If "diagnostic" or "prognostic" the values MUST be sent from the atmosphere model. For more information on how to send CO<sub>2</sub> from the data atmosphere model see the Section called *Running stand-alone CLM with transient historical CO<sub>2</sub> concentration* in Chapter 4.

### CLM\_FORCE\_COLDSTART

CLM\_FORCE\_COLDSTART when set to `on`, *requires* that your simulation do a cold start from arbitrary initial conditions. If this is NOT set, it will use an initial condition file if it can find an appropriate one, and otherwise do a cold start. CLM\_FORCE\_COLDSTART is a good way to ensure that you are doing a cold start if that is what you want to do.

### CLM\_PT1\_NAME

CLM\_PT1\_NAME is used *ONLY* for a pt1\_pt1 resolution simulation to set the name of the single-point files to use. To see a list of the valid resolutions do this:

```
> cd models/lnd/clm/doc
> ../../bld/build-namelist -res list
```

The output of the above command is:

```
build-namelist - valid values for res (Horizontal resolutions
Note: 0.5x0.5 and 0.33x0.33 are only used for CLM tools):
      Values: default 512x1024 128x256 64x128 48x96 32x64 8x16 94x192 0.23x0.31 \
0.47x0.63 0.9x1.25 1.9x2.5 2.5x3.33 4x5 10x15 5x5_amazon 1x1_tropicatl 1x1_camdenNJ \
1x1_vancouverCAN 1x1_mexicocityMEX 1x1_asphaltjungleNJ 1x1_brazil 1x1_urbanc_alpha 1x1_numaIA \
1x1_smallvilleIA 0.5x0.5 0.33x0.33
      Default = 1.9x2.5
      (NOTE: resolution and mask and other settings may influence what the default is)
```

the valid resolutions that can be used with CLM\_PT1\_NAME are the ones that have city or nation names such as: 5x5\_amazon, 1x1\_vancouverCAN 1x1\_mexicocityMEX, or 1x1\_brazil. The "1x1\_" prefix means the file is for a single-point, while "5x5\_" prefix means it's for a region of five points in latitude by five points in longitude. Both regional and single point datasets can be used for CLM\_PT1\_NAME. If you create your own datasets you can also use CLM\_PT1\_NAME along with CLM\_USRDAT\_NAME (documented below), setting CLM\_PT1\_NAME to the value in CLM\_USRDAT\_NAME so that your datasets are used rather than the standard ones.o

## CLM\_USRDATA\_NAME

CLM\_USRDATA\_NAME provides a way to enter your own datasets into the initial namelist setup at "**configure** -case". The files you create must be named with specific naming conventions outlined in: the Section called *Creating your own single-point/regional surface datasets* in Chapter 5. To see what the expected names of the files are, use the **queryDefaultNamelist.pl** to see what the names will need to be. For example if your CLM\_USRDATA\_NAME will be "1x1\_boulderCO", with a "navy" land-mask, constant simulation year range, for 1850, the following will list what your filenames should be:

```
> cd models/lnd/clm/bld
> queryDefaultNamelist.pl -usrname "1x1_boulderCO" -options \
mask=navy,sim_year=1850,sim_year_range="constant" -csmdata $CSDATA
```

An example of using CLM\_USRDATA\_NAME for a simulation is given in Example 5-5.

**Note:** See the the Section called *Precedence of Options* section for the precedence of this option relative to the others.

## Configure time User Namelist

CLM\_NAMELIST\_OPTS as described above allows you to set any extra namelist items you would like to appear in your namelist after first **configured**. However, it only allows you a single line to enter namelist items, and strings must be quoted with &apos; which is a bit awkward. If you have a long list of namelist items you want to set (such as a long list of history fields) a convenient way to do it is to create a **user\_nl\_clm** that contains just the list of namelist variables you want to add to your initial namelist. The **user\_nl\_clm** will only be used when **configure** is run, so if you change it after **configure** -- it won't change anything. The file needs to be in valid FORTRAN namelist format, and the **configure** step will abort if there are syntax errors. It merely needs to be named correctly **user\_nl\_clm** and placed in your case directory (where your other **env\_\*.xml** files are). The namelist name actually doesn't have to be valid, but all the variable names must be. Here's an example **user\_nl\_clm** namelist that sets a bunch of history file related items, to create output history files monthly, daily, every six and 1 hours.

### Example 1-1. Example **user\_nl\_clm** namelist file

```
&clmexp
hist_fincl2 = 'TG','TBOT','FIRE','FIRA','FLDS','FSDS',
               'FSR','FSA','FGEV','FSH','FGR','TSOI',
               'ERRSOI','BUILDHEAT','SABV','SABG',
               'FSDSVD','FSDSND','FSDSVI','FSDSNI',
               'FSRVD','FSRNND','FSRVI','FSRNI',
               'TSA','FCTR','FCEV','QBOT','RH2M','H2OSOI',
               'H2OSNO','SOILLIQ','SOILICE',
               'TSA_U','TSA_R',
               'TREFMNAV_U','TREFMNAV_R',
               'TREFMXAV_U','TREFMXAV_R',
               'TG_U','TG_R',
               'RH2M_U','RH2M_R',
               'QRUNOFF_U','QRUNOFF_R',
               'SoilAlpha_U',
               'Qanth',
               'SWup','LWup','URBAN_AC','URBAN_HEAT'
hist_fincl3 = 'TG:I','FSA:I','SWup:I','URBAN_AC:I','URBAN_HEAT:I',
```

```
'TG_U:I', 'TG_R:I',
hist_finc14 = 'TG', 'FSA', 'SWup', 'URBAN_AC', 'URBAN_HEAT'
hist_mfilt = 1, 30, 28, 24
hist_nhtfrq = 0, -24, -6, -1
/
```

**Note:** See the the Section called *Precedence of Options* section for the precedence of this option relative to the others.

**Note:** In the above example we use an invalid namelist name &clmexp -- but it works anyway because the CLM **build-namelist** knows the namelist that specific variable names belong to, and it puts them there.

Obviously, all of this would be difficult to put in the CLM\_NAMELIST\_OPTS variable, especially having to put ' around all the character strings. For more information on the namelist variables being set here and what they mean, see the section on CLM namelists below, as well as the namelist definition that gives details on each variable.

## Precedence of Options

Note: The precedence for setting the values of namelist variables with the different env\_conf options is (highest to lowest):

1. Namelist values set by specific command-line options, like, -d, -sim\_year (i.e. CLM\_BLDNML\_OPTS env\_conf variable)
2. Values set on the command-line using the -namelist option, (i.e. CLM\_NAMELIST\_OPTS env\_conf variable)
3. Values read from the file specified by -infile, (i.e. user\_nl\_clm file)
4. Datasets from the -clm\_usr\_name option, (i.e. CLM\_USRDATA\_NAME env\_conf variable)
5. Values set from a use-case scenario, e.g., -use\_case (i.e. CLM\_NML\_USE\_CASEenv\_conf variable)
6. Values from the namelist defaults file.

Thus a setting in CLM\_BLDNML\_OPTS will override a setting for the same thing given in a use case with CLM\_NML\_USE\_CASE. Likewise, a setting in CLM\_NAMELIST\_OPTS will override a setting in user\_nl\_clm.

## Setting Your Initial Conditions File

Especially with CLMCN starting from initial conditions is very important. Even with CLMSP it takes many simulation years to get the model fully spunup. There are a couple different ways to provide an initial condition file.

the Section called *Doing a hybrid simulation to provide initial conditions*

the Section called *Doing a branch simulation to provide initial conditions*

the Section called *Providing a finidat file in your user\_nl\_clmfile*

the Section called *Adding a finidat file to the XML database*

**Note:** Your initial condition file MUST agree with the surface dataset you are using to run the simulation. If the two files do NOT agree you will get a run-time about a mis-match in PFT weights, or in the number of PFT's or columns. To get around this you'll need to use the Section called *Using interpinic to interpolate initial conditions to different resolutions* in Chapter 2 to interpolate your initial condition dataset.

### Doing a hybrid simulation to provide initial conditions

The first option is to setup a hybrid simulation and give a RUN\_REFCASE and RUN\_REFDATE to specify the reference case simulation name to use. When you setup most cases, at the standard resolutions of "f09" or "f19" it will already do this for you. For example, if you run an "I2000CN" compset at "f09\_g16" resolution the following settings will already be done for you.

```
./xmlchange -file env_conf.xml -id RUN_TYPE      -val hybrid
./xmlchange -file env_conf.xml -id RUN_REFCASE -val I2000CN_f09_g16_c100503
./xmlchange -file env_conf.xml -id RUN_REFDATE -val 0001-01-01
./xmlchange -file env_conf.xml -id GET_REFCASE -val TRUE
```

Setting the GET\_REFCASE option to TRUE means it will copy the files from the:

`$DIN_LOC_ROOT/ccsm4_init/I2000CN_f09_g16_c100503/0001-01-01` directory. Note, that the RUN\_REFCASE and RUN\_REFDATE variables are expanded to get the directory name above. If you do NOT set GET\_REFCASE to TRUE then you will need to have placed the file in your run directory yourself. In either case, the file is expected to be named:

`$RUN_REFCASE.clm2.r.$RUN_REFDATE-00000.nc` with the variables expanded of course.

### Doing a branch simulation to provide initial conditions

The setup for running a branch simulation is essentially the same as for a hybrid. With the exception of setting RUN\_TYPE to branch rather than hybrid. A branch simulation runs the case essentially as restarting from it's place before to exactly reproduce it. While a hybrid simulation allows you to change namelist items, and use a different code base that may have fewer fields on it than a full restart file. The GET\_REFCASE works similarly for a branch case as for a hybrid.

## **Providing a finidat file in your user\_nl\_clm file**

Setting up a branch or hybrid simulation requires the initial condition file to follow a standard naming convention, and a standard input directory if you use the GET\_REFCASE option. If you want to name your file willy nilly and place it anywhere, you can set it in your user\_nl\_clm file as in this example.

```
&clm_inparm  
finidat      = '/glade/home/$USER/myinitdata/clmi_I1850CN_f09_g16_0182-01-01.c120329.nc'  
/
```

Note, if you provide an initial condition file -- you can NOT set CLM\_FORCE\_COLDSTART to TRUE.

## **Adding a finidat file to the XML database**

Like other datasets, if you want to use a given initial condition file to be used for all (or most of) your cases you'll want to put it in the XML database so it will be used by default. The initial condition files, are resolution dependent, and dependent on the number of PFT's and other variables such as GLC\_NEK or if irrigation is on or off. See Chapter 3 for more information on this.

## **Other noteworthy configuration items**

For running "I" cases there are several other noteworthy configuration items that you may want to work with. Most of these involve settings for the DATM, but one CCSM\_CO2\_PPMV applies to all models. If you are running an B, E, or F case that doesn't use the DATM obviously the DATM\_\* settings will not be used. All of the settings below are in your env\_conf.xml file

```
CCSM_CO2_PPMV  
CCSM_VOC  
DATM_MODE  
DATM_PRESAERO  
DATM_CLMNCEP_YR_ALIGN  
DATM_CLMNCEP_YR_START  
DATM_CLMNCEP_YR_END  
DATM_CPL_CASE  
DATM_CPL_YR_ALIGN  
DATM_CPL_YR_START  
DATM_CPL_YR_END
```

### **CCSM\_CO2\_PPMV**

CCSM\_CO2\_PPMV sets the mixing ratio of CO<sub>2</sub> in parts per million by volume for ALL CESM components to use. Note that most compsets already set this value to something reasonable. Also note that some compsets may tell the atmosphere model to override this value with either historic or ramped values. If the CCSM\_BGC variable is set to something other than "none" the atmosphere model will determine CO<sub>2</sub>, and CLM will listen and use what the atmosphere sends it. On the CLM side the namelist item co2\_type tells CLM to use the value sent from the atmosphere rather than a value set on its own namelist.

## CCSM\_VOC

CCSM\_VOC enables passing of the Volatile Organic Compounds (VOC) from CLM to the atmospheric model. This of course is only important if the atmosphere model is a fully active model that can use these fields in it's chemistry calculations.

## DATM\_MODE

DATM\_MODE sets the mode that the DATM model should run in this determines how data is handled as well as what the source of the data will be. Many of the modes are setup specifically to be used for ocean and/or sea-ice modeling. The modes that are designed for use by CLM are:

`CLM_QIAN`  
`CLM1PT`  
`CPLHIST3HrWx`

`CLM_QIAN` is for the standard mode of using global atmospheric data that was developed by Qian et. al. for CLM using NCEP data from 1948 to 2004. See the Section called *CLM\_QIAN mode and it's DATM settings* for more information on the DATM settings for `CLM_QIAN` mode. `CLM1PT` is for the special cases where we have single-point tower data for particular sites. Right now we only have data for three urban locations: MexicoCity Mexico, Vancouver Canada, and the urban-c alpha site. See the Section called *CLM1PT mode and it's DATM settings* for more information on the DATM settings for `CLM1PT` mode. `CPLHIST3HrWx` is for running with atmospheric forcing from a previous CESM simulation. See the Section called *CPLHIST3HrWx mode and it's DATM settings* for more information on the DATM settings for `CPLHIST3HrWx` mode.

## DATM\_PRESAERO

DATM\_PRESAERO sets the prescribed aerosol mode for the data atmosphere model. The list of valid options include:

`clim_1850` = constant year 1850 conditions  
`clim_2000` = constant year 2000 conditions  
`trans_1850-2000` = transient 1850 to year 2000 conditions  
`rcp2.6` = transient conditions for the rcp=2.6 W/m<sub>2</sub> future scenario  
`rcp4.5` = transient conditions for the rcp=4.5 W/m<sub>2</sub> future scenario  
`rcp6.0` = transient conditions for the rcp=6.0 W/m<sub>2</sub> future scenario  
`rcp8.5` = transient conditions for the rcp=8.5 W/m<sub>2</sub> future scenario  
`pt1_pt1` = read in single-point or regional datasets

## DATM\_CLMNCEP\_YR\_START

DATM\_CLMNCEP\_YR\_START sets the beginning year to cycle the atmospheric data over for the `CLM_QIAN` mode.

## DATM\_CLMNCEP\_YR\_END

DATM\_CLMNCEP\_YR\_END sets the ending year to cycle the atmospheric data over for the `CLM_QIAN` mode.

## DATM\_CLMNCEP\_YR\_ALIGN

DATM\_CLMNCEP\_YR\_START and DATM\_CLMNCEP\_YR\_END determine the range of years to cycle the atmospheric data over, and DATM\_CLMNCEP\_YR\_ALIGN determines which year in that range of years the simulation will start with.

#### DATM\_CPL\_CASE

DATM\_CPL\_CASE sets the casename to use for the CPLHIST3HrWx mode.

#### DATM\_CPL\_YR\_START

DATM\_CPL\_YR\_START sets the beginning year to cycle the atmospheric data over for the CPLHIST3HrWx mode.

#### DATM\_CPL\_YR\_END

DATM\_CPL\_YR\_END sets the ending year to cycle the atmospheric data over for the CPLHIST3HrWx mode.

#### DATM\_CPL\_YR\_ALIGN

DATM\_CPL\_YR\_START and DATM\_CPL\_YR\_END determine the range of years to cycle the atmospheric data over, and DATM\_CPL\_YR\_ALIGN determines which year in that range of years the simulation will start with.

## Downloading DATM Forcing Data

In Chapter One of the CESM User's Guide

([http://www.cesm.ucar.edu/models/cesm1.0/cesm/cesm\\_doc/book1.html](http://www.cesm.ucar.edu/models/cesm1.0/cesm/cesm_doc/book1.html)) there is a section on "Downloading input data". The normal process of setting up cases will use the "scripts/ccsm\_utils/Tools/check\_input\_data" script to retrieve data from the CESM subversion inputdata repository. However, the DATM forcing data is unique -- because it is large compared to the rest of the input data (56 Gbytes). Most of the data is stored in the directory set by the env\_run.xml variable DIN\_LOC\_ROOT\_CSMDATA. The CLM\_QIAN forcing data is in a (possibly) separate directory using the env\_run.xml variable DIN\_LOC\_ROOT\_CLMQIAN. In most cases this directory will be in the directory: atm/datm7/atm\_forcing.datm7.Qian.T62.c080727 under DIN\_LOC\_ROOT\_CSMDATA. On bluefire there is a separate path for the CLM\_QIAN forcing data. We have the full set of data available on a few of the machines we use: bluefire, jaguarpf, and edinburgh. As of October, 18th, 2011 we've uploaded the entire set of forcing data into the input data repository so now it can be treated like other input datasets and the check\_input\_data script can retrieve it for you. Previously only two years of data was available. You can also download the data from the Earth System Grid for other machines. See the Model Forcing Data ([http://www.cesm.ucar.edu/models/cesm1.0/clm\\_forcingdata\\_esg.html](http://www.cesm.ucar.edu/models/cesm1.0/clm/clm_forcingdata_esg.html)) link under the CLM Documentation Page (<http://www.cesm.ucar.edu/models/cesm1.0/clm>)

## Customizing via the template files

The final thing that the user may wish to do before **configure** is run is to edit the template files which determine the configuration and initial namelist. The variables in env\_conf.xml typically mean you will NOT have to edit the template. But, there are rare instances where it is useful to do so. Appendix A gives the details on how to do this. The template files are copied to your case directory and are available under Tools/Templates. The list of template files you might wish to edit are:

clm.cpl7.template

```
datm.cpl7.template  
cpl.template
```

## More information on the CLM configure script

The **configure** script defines the details of a clm configuration and summarizes it into a `config_cache.xml` file. The `config_cache.xml` will be placed in your case directory under `Buildconf/clmconf`. The `config_definition.xml` (`../../bld/config_files/config_definition.xml`) in `models/lnd/clm/bld/config_files` gives a definition of each CLM configuration item, it is viewable in a web-browser. Many of these items are things that you would NOT change, but looking through the list gives you the valid options, and a good description of each. Below we repeat the `config_definition.xml` files contents:

### Help on CLM configure

Coupling this with looking at the options to **configure** with "-help" as below will enable you to understand how to set the different options.

```
> cd models/lnd/clm/bld  
> configure -help
```

The output to the above command is as follows:

We've given details on how to use the options in `env_conf.xml` to interact with the CLM "**configure**" and "**build-namelist**" scripts, as well as giving a good understanding of how these scripts work and the options to them. In the next section we give further details on the CLM namelist. You could customize the namelist for these options after "**configure** -case" is run.

## 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 (../../bld/namelist\_files/namelist\_definition.xml)
2. Default values of each CLM Namelist Item (../../bld/namelist\_files/namelist\_defaults\_clm.xml)

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 (../../bld/namelist\_files/history\_fields.xml) documents all of the history fields available and gives the long-name and units for each.

## Definition of CLM history variables

Included in the table are the following pieces of information:

- Variable name.
- Long name description.
- units

**Table 1-1. CLM History Fields**

Name	Long-name	Units
A10TMIN	10-day running mean of min 2-m temperature	K
A5TMIN	5-day running mean of min 2-m temperature	K
ACTUAL_IMMOB	actual N immobilization	gN/m^2/s
AGDD	growing degree-days base 5C	K
AGNPP	aboveground NPP	gC/m^2/s
AGNPP	above-ground net primary production	gC/m2/s
ALBD	surface albedo (direct)	proportion
ALBGRD	ground albedo (direct)	proportion
ALBGRI	ground albedo (indirect)	proportion
ALBI	surface albedo (indirect)	proportion
ALLOC_PNOW	fraction of current allocation to display as new growth	proportion
ALPHA	alpha coefficient for VOC calc	non
ALPHAPSNSHA	shaded c13 fractionation	proportion
ALPHAPSNSUN	sunlit c13 fractionation	proportion
ANNAVG_T2M	annual average 2m air temperature	K

Name	Long-name	Units
ANNMAX_RETRANSN	annual max of retranslocated N pool	gN/m^2
ANNSUM_COUNTER	seconds since last annual accumulator turnover	s
ANNSUM_NPP	annual sum of NPP	gC/m^2/yr
ANNSUM_POTENTIAL_GPP	annual sum of potential GPP	gN/m^2/yr
ANN_FAREA_BURNED	annual total fractional area burned	proportion
AR	autotrophic respiration (MR + GR)	gC/m^2/s
AR	autotrophic respiration	gC/m2/s
AVAILC	C flux available for allocation	gC/m^2/s
AVAIL_RETRANSN	N flux available from retranslocation pool	gN/m^2/s
BCDEP	total BC deposition (dry+wet) from atmosphere	kg/m^2/s
BETA	coefficient of convective velocity	none
BGLFR	background litterfall rate	1/s
BGMOIST	moisture dependence	
BGNPP	belowground NPP	gC/m^2/s
BGNPP	below-ground net primary production	gC/m2/s
BGTEMP	temperature dependence	
BGTR	background transfer growth rate	1/s
BIOGENCO	biogenic CO flux	uGC/M2/H
BTRAN	transpiration beta factor	unitless
BUILDHEAT	heat flux from urban building interior to walls and roof	watt/m^2
C13_AGNPP	C13 aboveground NPP	gC13/m^2/s
C13_AR	C13 autotrophic respiration (MR + GR)	gC13/m^2/s
C13_BGNPP	C13 belowground NPP	gC13/m^2/s
C13_COL_CTRUNC	C13 column-level sink for C truncation	gC13/m^2
C13_COL_FIRE_CLOSS	C13 total column-level fire C loss	gC13/m^2/s
C13_CPOOL	C13 temporary photosynthate C pool	gC13/m^2
C13_CPOOL_DEADCROOT_GR	C13 dead coarse root growth respiration	gC13/m^2/s

Name	Long-name	Units
C13_CPOOL_DEADCROOT_STORAGE	C13 coarse root growth respiration to storage	gC13/m^2/s
C13_CPOOL_DEADSTEM_GR	C13 dead stem growth respiration	gC13/m^2/s
C13_CPOOL_DEADSTEM_STORAGE	C13 dead stem growth respiration to storage	gC13/m^2/s
C13_CPOOL_FROOT_GR	C13 fine root growth respiration	gC13/m^2/s
C13_CPOOL_FROOT_STORAGE	C13 fine root growth respiration to storage	gC13/m^2/s
C13_CPOOL_LEAF_GR	C13 leaf growth respiration	gC13/m^2/s
C13_CPOOL_LEAF_STORAGE	C13 leaf growth respiration to storage	gC13/m^2/s
C13_CPOOL_LIVECROOT_GR	C13 live coarse root growth respiration	gC13/m^2/s
C13_CPOOL_LIVECROOT_STORAGE	C13 live coarse root growth respiration to storage	gC13/m^2/s
C13_CPOOL_LIVESTEM_GR	C13 live stem growth respiration	gC13/m^2/s
C13_CPOOL_LIVESTEM_STORAGE	C13 live stem growth respiration to storage	gC13/m^2/s
C13_CPOOL_TO_DEADCROOTC	C13 allocation to dead coarse root C	gC13/m^2/s
C13_CPOOL_TO_DEADCROOTC_STORAGE	C13 dead coarse root C storage	gC13/m^2/s
C13_CPOOL_TO_DEADSTEMC	C13 allocation to dead stem C	gC13/m^2/s
C13_CPOOL_TO_DEADSTEMC_STORAGE	C13 allocation to dead stem C storage	gC13/m^2/s
C13_CPOOL_TO_FROOTC	C13 allocation to fine root C	gC13/m^2/s
C13_CPOOL_TO_FROOTC_STORAGE	C13 allocation to fine root C storage	gC13/m^2/s
C13_CPOOL_TO_GRESP_STORAGE	C13 allocation to growth respiration storage	gC13/m^2/s
C13_CPOOL_TO_LEAFC	C13 allocation to leaf C	gC13/m^2/s
C13_CPOOL_TO_LEAFC_STORAGE	C13 allocation to leaf C storage	gC13/m^2/s
C13_CPOOL_TO_LIVECROOTC	C13 allocation to live coarse root C	gC13/m^2/s
C13_CPOOL_TO_LIVECROOTC_STORAGE	C13 allocation to live coarse root C storage	gC13/m^2/s
C13_CPOOL_TO_LIVESTEMC	C13 allocation to live stem C	gC13/m^2/s

Name	Long-name	Units
C13_CPOOL_TO_LIVESTEMC_STORAGE	C13 Cpool to live stem C storage	gC13/m^2/s
C13_CURRENT_GR	C13 growth resp for new growth displayed in this timestep	gC13/m^2/s
C13_CWDC	C13 coarse woody debris C	gC13/m^2
C13_CWDC_TO_LITR2C	C13 decomp. of coarse woody debris C to litter 2 C	gC13/m^2/s
C13_CWDC_TO_LITR3C	C13 decomp. of coarse woody debris C to litter 3 C	gC13/m^2/s
C13_DEADCROOTC	C13 dead coarse root C	gC13/m^2
C13_DEADCROOTC_STORAGE	C13 dead coarse root C storage	gC13/m^2
C13_DEADCROOTC_STORAGE_TO_XFER	C13 dead coarse root C shift storage to transfer	gC13/m^2/s
C13_DEADCROOTC_XFER	C13 dead coarse root C transfer	gC13/m^2
C13_DEADCROOTC_XFER_TO_DEADCROOTC	C13 dead coarse root C growth from storage	gC13/m^2/s
C13_DEADSTEMC	C13 dead stem C	gC13/m^2
C13_DEADSTEMC_STORAGE	C13 dead stem C storage	gC13/m^2
C13_DEADSTEMC_STORAGE_TO_XFER	C13 dead stem C shift storage to transfer	gC13/m^2/s
C13_DEADSTEMC_XFER	C13 dead stem C transfer	gC13/m^2
C13_DEADSTEMC_XFER_TO_DEADSTEMC	C13 dead stem C growth from storage	gC13/m^2/s
C13_DISPVEGC	C13 displayed veg carbon, excluding storage and cpool	gC13/m^2
C13_DWT_CLOSS	C13 total carbon loss from land cover conversion	gC13/m^2/s
C13_DWT_CONV_CFLUX	C13 conversion C flux (immediate loss to atm)	gC13/m^2/s
C13_DWT_DEADCROOTC_TO_CWD	C13 coarse root to CWD due to landcover change	gC13/m^2/s
C13_DWT_FROOTC_TO_LITTER	C13 fine root to litter due to landcover change	gC13/m^2/s
C13_DWT_FROOTC_TO_LITTER2	C13 fine root to litter due to landcover change	gC13/m^2/s
C13_DWT_FROOTC_TO_LITTER3	C13 fine root to litter due to landcover change	gC13/m^2/s
C13_DWT_LIVECROOTC_TO_CWD	C13 coarse root to CWD due to landcover change	gC13/m^2/s

Name	Long-name	Units
C13_DWT_PROD100C_GAIN	C13 addition to 100-yr wood product pool	gC13/m^2/s
C13_DWT_PROD10C_GAIN	C13 addition to 10-yr wood product pool	gC13/m^2/s
C13_DWT_SEEDC_TO_DEADSTEM	Seed source to PFT-level deadstem	gC13/m^2/s
C13_DWT_SEEDC_TO_LEAF	C13 seed source to PFT-level leaf	gC13/m^2/s
C13_ER	C13 total ecosystem respiration, autotrophic + heterotrophic	gC13/m^2/s
C13_FROOTC	C13 fine root C	gC13/m^2
C13_FROOTC_STORAGE	C13 fine root C storage	gC13/m^2
C13_FROOTC_STORAGE_TO_XFER	fine root C shift storage to transfer	gC13/m^2/s
C13_FROOTC_TO_LITR1C	C13 fine root C litterfall to litter 1 C	gC13/m^2/s
C13_FROOTC_TO_LITR2C	C13 fine root C litterfall to litter 2 C	gC13/m^2/s
C13_FROOTC_TO_LITR3C	C13 fine root C litterfall to litter 3 C	gC13/m^2/s
C13_FROOTC_TO_LITTER	C13 fine root C litterfall	gC13/m^2/s
C13_FROOTC_XFER	C13 fine root C transfer	gC13/m^2
C13_FROOTC_XFER_TO_FROOTBG	fine root C growth from storage	gC13/m^2/s
C13_FROOT_MR	C13 fine root maintenance respiration	gC13/m^2/s
C13_GPP	C13 gross primary production	gC13/m^2/s
C13_GR	C13 total growth respiration	gC13/m^2/s
C13_GRESP_STORAGE	C13 growth respiration storage	gC13/m^2
C13_GRESP_STORAGE_TO_XFER	growth respiration shift storage to transfer	gC13/m^2/s
C13_GRESP_XFER	C13 growth respiration transfer	gC13/m^2
C13_HR	C13 total heterotrophic respiration	gC13/m^2/s
C13_LEAFC	C13 leaf C	gC13/m^2
C13_LEAFC_STORAGE	C13 leaf C storage	gC13/m^2
C13_LEAFC_STORAGE_TO_XFER	leaf C shift storage to transfer	gC13/m^2/s
C13_LEAFC_TO_LITR1C	C13 leaf C litterfall to litter 1 C	gC13/m^2/s
C13_LEAFC_TO_LITR2C	C13 leaf C litterfall to litter 2 C	gC13/m^2/s
C13_LEAFC_TO_LITR3C	C13 leaf C litterfall to litter 3 C	gC13/m^2/s

Name	Long-name	Units
<b>C13_LEAFC_TO_LITTER</b>	C13 leaf C litterfall	gC13/m^2/s
<b>C13_LEAFC_XFER</b>	C13 leaf C transfer	gC13/m^2
<b>C13_LEAFC_XFER_TO_LEAF</b>	C13 leaf C growth from storage	gC13/m^2/s
<b>C13_LEAF_MR</b>	C13 leaf maintenance respiration	gC13/m^2/s
<b>C13_LITFALL</b>	C13 litterfall (leaves and fine roots)	gC13/m^2/s
<b>C13_LITFIRE</b>	C13 litter fire losses	gC13/m^2/s
<b>C13_LITHR</b>	C13 litter heterotrophic respiration	gC13/m^2/s
<b>C13_LITR1C</b>	C13 litter labile C	gC13/m^2
<b>C13_LITR1C_TO_SOIL1C</b>	C13 decomp. of litter 1 C to SOM 1 C	gC13/m^2/s
<b>C13_LITR1_HR</b>	C13 het. resp. from litter 1 C	gC13/m^2/s
<b>C13_LITR2C</b>	C13 litter cellulose C	gC13/m^2
<b>C13_LITR2C_TO_SOIL2C</b>	C13 decomp. of litter 2 C to SOM 2 C	gC13/m^2/s
<b>C13_LITR2_HR</b>	C13 het. resp. from litter 2 C	gC13/m^2/s
<b>C13_LITR3C</b>	C13 litter lignin C	gC13/m^2
<b>C13_LITR3C_TO_SOIL3C</b>	C13 decomp. of litter 3 C to SOM 3 C	gC13/m^2/s
<b>C13_LITR3_HR</b>	C13 het. resp. from litter 3 C	gC13/m^2/s
<b>C13_LIVECROOTC</b>	C13 live coarse root C	gC13/m^2
<b>C13_LIVECROOTC_STORAGE</b>	C13 live coarse root C storage	gC13/m^2
<b>C13_LIVECROOTC_STORAGE_TO_XFER</b>	C13 live coarse root C shift storage to transfer	gC13/m^2/s
<b>C13_LIVECROOTC_TO_DEADCROOTC</b>	C13 live coarse root C turnover	gC13/m^2/s
<b>C13_LIVECROOTC_XFER</b>	C13 live coarse root C transfer	gC13/m^2
<b>C13_LIVECROOTC_XFER_TO_LIVECROOTC</b>	C13 live coarse root C growth from storage	gC13/m^2/s
<b>C13_LIVECROOT_MR</b>	C13 live coarse root maintenance respiration	gC13/m^2/s
<b>C13_LIVESTEMC</b>	C13 live stem C	gC13/m^2
<b>C13_LIVESTEMC_STORAGE</b>	C13 live stem C storage	gC13/m^2
<b>C13_LIVESTEMC_STORAGE_TO_XFER</b>	C13 live stem C shift storage to transfer	gC13/m^2/s

Name	Long-name	Units
C13_LIVESTEMC_TO_DEADSTEMC	live stem C turnover	gC13/m^2/s
C13_LIVESTEMC_XFER	C13 live stem C transfer	gC13/m^2
C13_LIVESTEMC_XFER_TO_LIVESTEMC	LIVESTEMC growth from storage	gC13/m^2/s
C13_LIVESTEM_MR	C13 live stem maintenance respiration	gC13/m^2/s
C13_MR	C13 maintenance respiration	gC13/m^2/s
C13_M_CWDC_TO_FIRE	C13 coarse woody debris C fire loss	gC13/m^2/s
C13_M_DEADCROOTC_STORAGE_TO_FIRE	dead coarse root C storage fire loss	gC13/m^2/s
C13_M_DEADCROOTC_STORAGE_TO_LITTER1C	dead coarse root C storage mortality to litter 1 C	gC13/m^2/s
C13_M_DEADCROOTC_STORAGE_TO_LITTER1C	dead coarse root C storage mortality	gC13/m^2/s
C13_M_DEADCROOTC_TO_CWD	dead coarse root C mortality to coarse woody debris C	gC13/m^2/s
C13_M_DEADCROOTC_TO_CWD	dead coarse root C to coarse woody debris C by fire	gC13/m^2/s
C13_M_DEADCROOTC_TO_FIRE	dead coarse root C fire loss	gC13/m^2/s
C13_M_DEADCROOTC_TO_LITTER	dead coarse root C mortality	gC13/m^2/s
C13_M_DEADCROOTC_TO_LITTER	dead coarse root C fire mortality to litter	gC13/m^2/s
C13_M_DEADCROOTC_XFER_TO_FIRE	dead coarse root C transfer fire loss	gC13/m^2/s
C13_M_DEADCROOTC_XFER_TO_LITTER1C	dead coarse root C transfer mortality to litter 1 C	gC13/m^2/s
C13_M_DEADCROOTC_XFER_TO_LITTER1C	dead coarse root C transfer mortality	gC13/m^2/s
C13_M_DEADSTEMC_STORAGE_TO_FIRE	dead stem C storage fire loss	gC13/m^2/s
C13_M_DEADSTEMC_STORAGE_TO_LITTER1C	dead stem C storage mortality to litter 1 C	gC13/m^2/s
C13_M_DEADSTEMC_STORAGE_TO_LITTER	dead stem C storage mortality	gC13/m^2/s
C13_M_DEADSTEMC_TO_CWD	dead stem C mortality to coarse woody debris C	gC13/m^2/s
C13_M_DEADSTEMC_TO_CWD	dead stem C to coarse woody debris C by fire	gC13/m^2/s

Name	Long-name	Units
C13_M_DEADSTEMC_TO_FIRE	13 dead stem C fire loss	gC13/m^2/s
C13_M_DEADSTEMC_TO_LITTER	dead stem C mortality	gC13/m^2/s
C13_M_DEADSTEMC_TO_LITTER	dead stem C fire mortality to litter	gC13/m^2/s
C13_M_DEADSTEMC_XFER	stem C transfer fire loss	gC13/m^2/s
C13_M_DEADSTEMC_XFER	stem C transfer mortality to litter 1 C	gC13/m^2/s
C13_M_DEADSTEMC_XFER	stem C transfer mortality	gC13/m^2/s
C13_M_FROOTC_STORAGE	root C storage fire loss	gC13/m^2/s
C13_M_FROOTC_STORAGE	root C storage mortality to litter 1 C	gC13/m^2/s
C13_M_FROOTC_STORAGE	root C storage mortality	gC13/m^2/s
C13_M_FROOTC_TO_FIRE	C13 fine root C fire loss	gC13/m^2/s
C13_M_FROOTC_TO_LITTER	C13 fine root C mortality to litter 1 C	gC13/m^2/s
C13_M_FROOTC_TO_LITTER	C13 fine root C mortality to litter 2 C	gC13/m^2/s
C13_M_FROOTC_TO_LITTER	C13 fine root C mortality to litter 3 C	gC13/m^2/s
C13_M_FROOTC_TO_LITTER	C13 fine root C mortality	gC13/m^2/s
C13_M_FROOTC_XFER_TO_FIRE	fine root C transfer fire loss	gC13/m^2/s
C13_M_FROOTC_XFER_TO_LITTER	fine root C transfer mortality to litter 1 C	gC13/m^2/s
C13_M_FROOTC_XFER_TO_LITTER	fine root C transfer mortality	gC13/m^2/s
C13_M_GRESP_STORAGE_TO_FIRE	growth respiration storage fire loss	gC13/m^2/s
C13_M_GRESP_STORAGE_TO_LITTER	growth respiration storage mortality to litter 1 C	gC13/m^2/s
C13_M_GRESP_STORAGE_TO_LITTER	growth respiration storage mortality	gC13/m^2/s
C13_M_GRESP_XFER_TO_FIRE	3 growth respiration transfer fire loss	gC13/m^2/s

Name	Long-name	Units
C13_M_GRESP_XFER_TO_LITTER	growth respiration transfer mortality to litter 1 C	gC13/m^2/s
C13_M_GRESP_XFER_TO_LITTER	growth respiration transfer mortality	gC13/m^2/s
C13_M_LEAFC_STORAGE_TO_FIRE	leaf C storage fire loss	gC13/m^2/s
C13_M_LEAFC_STORAGE_TO_LITTER	storage mortality to litter 1 C	gC13/m^2/s
C13_M_LEAFC_STORAGE_TO_LITTER	storage mortality	gC13/m^2/s
C13_M_LEAFC_TO_FIRE	C13 leaf C fire loss	gC13/m^2/s
C13_M_LEAFC_TO_LITR1C	C13 leaf C mortality to litter 1 C	gC13/m^2/s
C13_M_LEAFC_TO_LITR2C	C13 leaf C mortality to litter 2 C	gC13/m^2/s
C13_M_LEAFC_TO_LITR3C	C13 leaf C mortality to litter 3 C	gC13/m^2/s
C13_M_LEAFC_TO_LITTER	C13 leaf C mortality	gC13/m^2/s
C13_M_LEAFC_XFER_TO_FIRE	leaf C transfer fire loss	gC13/m^2/s
C13_M_LEAFC_XFER_TO_LITR1C	leaf C transfer mortality to litter 1 C	gC13/m^2/s
C13_M_LEAFC_XFER_TO_LITTER	leaf C transfer mortality	gC13/m^2/s
C13_M_LITR1C_TO_FIRE	C13 litter 1 C fire loss	gC13/m^2/s
C13_M_LITR2C_TO_FIRE	C13 litter 2 C fire loss	gC13/m^2/s
C13_M_LITR3C_TO_FIRE	C13 litter 3 C fire loss	gC13/m^2/s
C13_M_LIVECROOTC_STORAGE_TO_FIRE	root C storage fire loss	gC13/m^2/s
C13_M_LIVECROOTC_STORAGE_TO_LITR1C	root C storage mortality to litter 1 C	gC13/m^2/s
C13_M_LIVECROOTC_STORAGE_TO_LITTER	root C storage mortality	gC13/m^2/s
C13_M_LIVECROOTC_TO_CWD	coarse root C mortality to coarse woody debris C	gC13/m^2/s
C13_M_LIVECROOTC_TO_FIRE	3 live coarse root C fire loss	gC13/m^2/s
C13_M_LIVECROOTC_TO_LITTER	live coarse root C mortality	gC13/m^2/s
C13_M_LIVECROOTC_XFER_TO_FIRE	coarse root C transfer fire loss	gC13/m^2/s
C13_M_LIVECROOTC_XFER_TO_LITR1C	coarse root C transfer mortality to litter 1 C	gC13/m^2/s

Name	Long-name	Units
C13_M_LIVECROOTC_XFER	TO_LITTER root C transfer mortality	gC13/m^2/s
C13_M_LIVESTEMC_STORAGE_310eFIRE	storage fire loss	gC13/m^2/s
C13_M_LIVESTEMC_STORAGE_310eLITTER	storage mortality to litter 1 C	gC13/m^2/s
C13_M_LIVESTEMC_STORAGE_310eLITTER	storage mortality	gC13/m^2/s
C13_M_LIVESTEMC_TO_CW	10C3 live stem C mortality to coarse woody debris C	gC13/m^2/s
C13_M_LIVESTEMC_TO_FIRE	C13 live stem C fire loss	gC13/m^2/s
C13_M_LIVESTEMC_TO_LITTER	live stem C mortality	gC13/m^2/s
C13_M_LIVESTEMC_XFER_TO_FIRE	stem C transfer fire loss	gC13/m^2/s
C13_M_LIVESTEMC_XFER_TO_LITTER1C	m C transfer mortality to litter 1 C	gC13/m^2/s
C13_M_LIVESTEMC_XFER_TO_LITTER1n	n C transfer mortality	gC13/m^2/s
C13_NEE	C13 net ecosystem exchange of carbon, includes fire flux, positive for source	gC13/m^2/s
C13_NEP	C13 net ecosystem production, excludes fire flux, positive for sink	gC13/m^2/s
C13_NPP	C13 net primary production	gC13/m^2/s
C13_PFT_CTRUNC	C13 pft-level sink for C truncation	gC13/m^2
C13_PFT_FIRE_CLOSS	C13 total pft-level fire C loss	gC13/m^2/s
C13_PROD100C	C13 100-yr wood product C	gC13/m^2
C13_PROD100C_LOSS	C13 loss from 100-yr wood product pool	gC13/m^2/s
C13_PROD10C	C13 10-yr wood product C	gC13/m^2
C13_PROD10C_LOSS	C13 loss from 10-yr wood product pool	gC13/m^2/s
C13_PRODUCT_CLOSS	C13 total carbon loss from wood product pools	gC13/m^2/s
C13_PSNSHA	C13 shaded leaf photosynthesis	umolCO2/m^2/s
C13_PSNSHADE_TO_CPOOL	C13 C fixation from shaded canopy	gC13/m^2/s
C13_PSNSUN	C13 sunlit leaf photosynthesis	umolCO2/m^2/s

Name	Long-name	Units
<b>C13_PSNSUN_TO_CPOOL</b>	C13 C fixation from sunlit canopy	gC13/m^2/s
<b>C13_RR</b>	C13 root respiration (fine root MR + total root GR)	gC13/m^2/s
<b>C13_SEEDC</b>	C13 pool for seeding new PFTs	gC13/m^2
<b>C13_SOIL1C</b>	C13 soil organic matter C (fast pool)	gC13/m^2
<b>C13_SOIL1C_TO_SOIL2C</b>	C13 decomp. of SOM 1 C to SOM 2 C	gC13/m^2/s
<b>C13_SOIL1_HR</b>	C13 het. resp. from SOM 1 C	gC13/m^2/s
<b>C13_SOIL2C</b>	C13 soil organic matter C (medium pool)	gC13/m^2
<b>C13_SOIL2C_TO_SOIL3C</b>	C13 decomp. of SOM 2 C to SOM 3 C	gC13/m^2/s
<b>C13_SOIL2_HR</b>	C13 het. resp. from SOM 2 C	gC13/m^2/s
<b>C13_SOIL3C</b>	C13 soil organic matter C (slow pool)	gC13/m^2
<b>C13_SOIL3C_TO_SOIL4C</b>	C13 decomp. of SOM 3 C to SOM 4 C	gC13/m^2/s
<b>C13_SOIL3_HR</b>	C13 het. resp. from SOM 3 C	gC13/m^2/s
<b>C13_SOIL4C</b>	C13 soil organic matter C (slowest pool)	gC13/m^2
<b>C13_SOIL4_HR</b>	C13 het. resp. from SOM 4 C	gC13/m^2/s
<b>C13_SOMFIRE</b>	C13 soil organic matter fire losses	gC13/m^2/s
<b>C13_SOMHR</b>	C13 soil organic matter heterotrophic respiration	gC13/m^2/s
<b>C13_SR</b>	C13 total soil respiration (HR + root resp)	gC13/m^2/s
<b>C13_STORAGE_GR</b>	C13 growth resp for growth sent to storage for later display	gC13/m^2/s
<b>C13_STORVEGC</b>	C13 stored vegetation carbon, excluding cpool	gC13/m^2
<b>C13_TOTCOLC</b>	C13 total column carbon, incl veg and cpool	gC13/m^2
<b>C13_TOTECOSYSC</b>	C13 total ecosystem carbon, incl veg but excl cpool	gC13/m^2
<b>C13_TOTFFIRE</b>	C13 total ecosystem fire losses	gC13/m^2/s
<b>C13_TOTLITC</b>	C13 total litter carbon	gC13/m^2
<b>C13_TOTPFTC</b>	C13 total pft-level carbon, including cpool	gC13/m^2

Name	Long-name	Units
<b>C13_TOTPRODC</b>	C13 total wood product C	gC13/m^2
<b>C13_TOTSOMC</b>	C13 total soil organic matter carbon	gC13/m^2
<b>C13_TOTVEGC</b>	C13 total vegetation carbon, excluding cpool	gC13/m^2
<b>C13_TRANSFER_DEADCROOT</b>	<b>T1GR</b> ead coarse root growth respiration from storage	gC13/m^2/s
<b>C13_TRANSFER_DEADSTEM</b>	<b>GR</b> ead stem growth respiration from storage	gC13/m^2/s
<b>C13_TRANSFER_FROOT_GR</b>	C13 fine root growth respiration from storage	gC13/m^2/s
<b>C13_TRANSFER_GR</b>	C13 growth resp for transfer growth displayed in this timestep	gC13/m^2/s
<b>C13_TRANSFER_LEAF_GR</b>	C13 leaf growth respiration from storage	gC13/m^2/s
<b>C13_TRANSFER_LIVECROOT</b>	<b>TGR</b> ive coarse root growth respiration from storage	gC13/m^2/s
<b>C13_TRANSFER_LIVESTEM</b>	<b>GR</b> ive stem growth respiration from storage	gC13/m^2/s
<b>C13_VEGFIRE</b>	C13 pft-level fire loss	gC13/m^2/s
<b>C13_XSMRPOOL</b>	C13 temporary photosynthate C pool	gC13/m^2
<b>CANNAVG_T2M</b>	annual average of 2m air temperature	K
<b>CANNSUM_NPP</b>	annual sum of column-level NPP	gC/m^2/s
<b>CFLUX</b>	total Carbon flux	g/m2/s
<b>CGRND</b>	deriv. of soil energy flux wrt to soil temp	W/m^2/K
<b>CGRNDL</b>	deriv. of soil latent heat flux wrt soil temp	W/m^2/K
<b>CGRNDS</b>	deriv. of soil sensible heat flux wrt soil temp	W/m^2/K
<b>CISHA</b>	shaded intracellular CO2	Pa
<b>CISUN</b>	sunlit intracellular CO2	Pa
<b>CLOSS</b>	Amt. of Carbon lost by pool	g/m2/s
<b>CO2FLUX</b>	net CO2 flux	g/m2/s
<b>COL_CTRUNC</b>	column-level sink for C truncation	gC/m^2
<b>COL_FIRE_CLOSS</b>	total column-level fire C loss	gC/m^2/s
<b>COL_FIRE_NLOSS</b>	total column-level fire N loss	gN/m^2/s

Name	Long-name	Units
COL_NTRUNC	column-level sink for N truncation	gN/m^2
COSZEN	cosine of solar zenith angle	none
CP	cp coefficient for VOC calc	non
CPOOL	temporary photosynthate C pool	gC/m^2
CPOOL_DEADCROOT_GR	dead coarse root growth respiration	gC/m^2/s
CPOOL_DEADCROOT_STORAGE_GR	dead coarse root growth respiration to storage	gC/m^2/s
CPOOL_DEADSTEM_GR	dead stem growth respiration	gC/m^2/s
CPOOL_DEADSTEM_STORAGE_GR	dead stem growth respiration to storage	gC/m^2/s
CPOOL_FROOT_GR	fine root growth respiration	gC/m^2/s
CPOOL_FROOT_STORAGE_GR	fine root growth respiration to storage	gC/m^2/s
CPOOL_LEAF_GR	leaf growth respiration	gC/m^2/s
CPOOL_LEAF_STORAGE_GR	leaf growth respiration to storage	gC/m^2/s
CPOOL_LIVECROOT_GR	live coarse root growth respiration	gC/m^2/s
CPOOL_LIVECROOT_STORAGE_GR	live coarse root growth respiration to storage	gC/m^2/s
CPOOL_LIVESTEM_GR	live stem growth respiration	gC/m^2/s
CPOOL_LIVESTEM_STORAGE_GR	live stem growth respiration to storage	gC/m^2/s
CPOOL_TO_DEADCROOTC	allocation to dead coarse root C	gC/m^2/s
CPOOL_TO_DEADCROOTC_STORAGE	allocation to dead coarse root C storage	gC/m^2/s
CPOOL_TO_DEADSTEMC	allocation to dead stem C	gC/m^2/s
CPOOL_TO_DEADSTEMC_STORAGE	allocation to dead stem C storage	gC/m^2/s
CPOOL_TO_FROOTC	allocation to fine root C	gC/m^2/s
CPOOL_TO_FROOTC_STORAGE	allocation to fine root C storage	gC/m^2/s
CPOOL_TO_GRESP_STORAGE	allocation to growth respiration storage	gC/m^2/s
CPOOL_TO_LEAFC	allocation to leaf C	gC/m^2/s
CPOOL_TO_LEAFC_STORAGE	allocation to leaf C storage	gC/m^2/s
CPOOL_TO_LIVECROOTC	allocation to live coarse root C	gC/m^2/s

Name	Long-name	Units
CPOOL_TO_LIVECROOTC_STORAGE	allocation to live coarse root C storage	gC/m^2/s
CPOOL_TO_LIVESTEMC	allocation to live stem C	gC/m^2/s
CPOOL_TO_LIVESTEMC_STORAGE	allocation to live stem C storage	gC/m^2/s
CTRANS	Amt. of Carbon transferred out of pool types	g/m2/s
CURRENT_GR	growth resp for new growth displayed in this timestep	gC/m^2/s
CWDC	coarse woody debris C	gC/m^2
CWDC	coarse woody debris C	gC/m2
CWDC_HR	coarse woody debris C heterotrophic respiration	gC/m^2/s
CWDC_HR	coarse woody debris heterotrophic respiration	gC/m2/s
CWDC_LOSS	coarse woody debris C loss	gC/m^2/s
CWDC_LOSS	coarse woody debris C loss	gC/m2/s
CWDC_TO_LITR2C	decomp. of coarse woody debris C to litter 2 C	gC/m^2/s
CWDC_TO_LITR3C	decomp. of coarse woody debris C to litter 3 C	gC/m^2/s
CWDN	coarse woody debris N	gN/m^2
CWDN_TO_LITR2N	decomp. of coarse woody debris N to litter 2 N	gN/m^2/s
CWDN_TO_LITR3N	decomp. of coarse woody debris N to litter 3 N	gN/m^2/s
C_ALLOMETRY	C allocation index	none
DAYL	daylength	s
DAYS_ACTIVE	number of days since last dormancy	days
DEADCROOTC	dead coarse root C	gC/m^2
DEADCROOTC_STORAGE	dead coarse root C storage	gC/m^2
DEADCROOTC_STORAGE_TO_XFER	dead coarse root C shift storage to transfer	gC/m^2/s
DEADCROOTC_XFER	dead coarse root C transfer	gC/m^2
DEADCROOTC_XFER_TO_DEADCROOTC	dead coarse root C growth from storage	gC/m^2/s
DEADCROOTN	dead coarse root N	gN/m^2
DEADCROOTN_STORAGE	dead coarse root N storage	gN/m^2
DEADCROOTN_STORAGE_TO_XFER	dead coarse root N shift storage to transfer	gN/m^2/s

Name	Long-name	Units
<b>DEADCROOTN_XFER</b>	dead coarse root N transfer	gN/m^2
<b>DEADCROOTN_TO_DEADCROOTN</b>	dead root N growth from storage	gN/m^2/s
<b>DEADSTEMC</b>	dead stem C	gC/m^2
<b>DEADSTEMC_STORAGE</b>	dead stem C storage	gC/m^2
<b>DEADSTEMC_STORAGE_TO_DEADSTEMC_XFER</b>	dead stem C shift storage to transfer	gC/m^2/s
<b>DEADSTEMC_XFER</b>	dead stem C transfer	gC/m^2
<b>DEADSTEMC_XFER_TO_DEADSTEMCC</b>	dead stem C growth from storage	gC/m^2/s
<b>DEADSTEMN</b>	dead stem N	gN/m^2
<b>DEADSTEMN_STORAGE</b>	dead stem N storage	gN/m^2
<b>DEADSTEMN_STORAGE_TO_DEADSTEMN_XFER</b>	dead stem N shift storage to transfer	gN/m^2/s
<b>DEADSTEMN_XFER</b>	dead stem N transfer	gN/m^2
<b>DEADSTEMN_XFER_TO_DEADSTEMNN</b>	dead stem N growth from storage	gN/m^2/s
<b>DECL</b>	solar declination angle	radians
<b>DEGDAY</b>	Accumulated degree days	deg C
<b>DENIT</b>	total rate of denitrification	gN/m^2/s
<b>DEWMX</b>	Maximum allowed dew	mm
<b>DGNETDT</b>	derivative of net ground heat flux wrt soil temp	W/m^2/K
<b>DISPLA</b>	displacement height	m
<b>DISPVEGC</b>	displayed veg carbon, excluding storage and cpool	gC/m^2
<b>DISPVEGN</b>	displayed vegetation nitrogen	gN/m^2
<b>DLRAD</b>	downward longwave radiation below the canopy	W/m^2
<b>DORMANT_FLAG</b>	dormancy flag	none
<b>DOWNREG</b>	fractional reduction in GPP due to N limitation	proportion
<b>DPVLTRB1</b>	turbulent deposition velocity 1	m/s
<b>DPVLTRB2</b>	turbulent deposition velocity 2	m/s
<b>DPVLTRB3</b>	turbulent deposition velocity 3	m/s
<b>DPVLTRB4</b>	turbulent deposition velocity 4	m/s
<b>DSTDEP</b>	total dust deposition (dry+wet) from atmosphere	kg/m^2/s
<b>DSTFLXT</b>	total surface dust emission	kg/m^2/s
<b>DT_VEG</b>	change in t_veg, last iteration	K

Name	Long-name	Units
DVOLRDT_LND	RTM land change in storage: LIQ	mm/s
DVOLRDT_LND_ICE	RTM land change in storage: ICE	mm/s
DVOLRDT_OCN	RTM ocean change of storage: LIQ	mm/s
DVOLRDT_OCN_ICE	RTM ocean change of storage: ICE	mm/s
DWT_CLOSS	total carbon loss from land cover conversion	gC/m^2/s
DWT_CONV_CFLUX	conversion C flux (immediate loss to atm)	gC/m^2/s
DWT_CONV_NFLUX	conversion N flux (immediate loss to atm)	gN/m^2/s
DWT_DEADCROOTC_TO_CWD	dead coarse root to CWD due to landcover change	gC/m^2/s
DWT_DEADCROOTN_TO_CWD	dead coarse root to CWD due to landcover change	gN/m^2/s
DWT_FROOTC_TO_LITR1C	fine root to litter due to landcover change	gC/m^2/s
DWT_FROOTC_TO_LITR2C	fine root to litter due to landcover change	gC/m^2/s
DWT_FROOTC_TO_LITR3C	fine root to litter due to landcover change	gC/m^2/s
DWT_FROOTN_TO_LITR1N	fine root to litter due to landcover change	gN/m^2/s
DWT_FROOTN_TO_LITR2N	fine root to litter due to landcover change	gN/m^2/s
DWT_FROOTN_TO_LITR3N	fine root to litter due to landcover change	gN/m^2/s
DWT_LIVECROOTC_TO_CWD	live coarse root to CWD due to landcover change	gC/m^2/s
DWT_LIVECROOTN_TO_CWD	live coarse root to CWD due to landcover change	gN/m^2/s
DWT_NLOSS	total nitrogen loss from landcover conversion	gN/m^2/s
DWT_PROD100C_GAIN	landcover change-driven addition to 100-yr wood product pool	gC/m^2/s
DWT_PROD100N_GAIN	addition to 100-yr wood product pool	gN/m^2/s
DWT_PROD10C_GAIN	landcover change-driven addition to 10-yr wood product pool	gC/m^2/s

Name	Long-name	Units
DWT_PROD10N_GAIN	addition to 10-yr wood product pool	gN/m^2/s
DWT_SEEDC_TO_DEADSTEM	seed source to PFT-level deadstem	gC/m^2/s
DWT_SEEDC_TO_LEAF	seed source to PFT-level leaf	gC/m^2/s
DWT_SEEDN_TO_DEADSTEM	seed source to PFT-level deadstem	gN/m^2/s
DWT_SEEDN_TO_LEAF	seed source to PFT-level leaf	gN/m^2/s
EFF_KID	effective extinction coefficient for indirect from direct	none
EFF_KII	effective extinction coefficient for indirect from indirect	none
EFF_POROSITY	effective porosity = porosity - vol_ice	proportion
EFLX_DYNBAL	dynamic land cover change conversion energy flux	W/m^2
EFLX_GNET	net heat flux into ground	W/m^2
EFLX_LH_TOT	total latent heat flux [+ to atm]	W/m^2
EFLX_LH_TOT_R	Rural total evaporation	watt/m^2
EFLX_LH_TOT_U	Urban total evaporation	watt/m^2
EFLX_SOIL_GRND	soil heat flux [+ into soil]	W/m^2
ELAI	exposed one-sided leaf area index	m^2/m^2
EMG	ground emissivity	proportion
EMV	vegetation emissivity	proportion
EOPT	Eopt coefficient for VOC calc	non
ER	total ecosystem respiration, autotrophic + heterotrophic	gC/m^2/s
ERRH2O	total water conservation error	mm
ERRSEB	surface energy conservation error	watt/m^2
ERRSOI	soil/lake energy conservation error	watt/m^2
ERRSOL	solar radiation conservation error	watt/m^2
ESAI	exposed one-sided stem area index	m^2/m^2
EXCESSC	excess carbon	g/m2/timestep
EXCESS_CFLUX	C flux not allocated due to downregulation	gC/m^2/s

Name	Long-name	Units
<b>FABD</b>	flux absorbed by veg per unit direct flux	proportion
<b>FABI</b>	flux absorbed by veg per unit indirect flux	proportion
<b>FAREA_BURNED</b>	timestep fractional area burned	proportion
<b>FCEV</b>	canopy evaporation	watt/m <sup>2</sup>
<b>FCOV</b>	fractional impermeable area	unitless
<b>FCTR</b>	canopy transpiration	watt/m <sup>2</sup>
<b>FDRY</b>	fraction of foliage that is green and dry	proportion
<b>FGEV</b>	ground evaporation	watt/m <sup>2</sup>
<b>FGR</b>	heat flux into soil/snow including snow melt	watt/m <sup>2</sup>
<b>FGR12</b>	heat flux between soil layers 1 and 2	watt/m <sup>2</sup>
<b>FGR_R</b>	Rural heat flux into soil/snow including snow melt	watt/m <sup>2</sup>
<b>FGR_U</b>	Urban heat flux into soil/snow including snow melt	watt/m <sup>2</sup>
<b>FIRA</b>	net infrared (longwave) radiation	watt/m <sup>2</sup>
<b>FIRA_R</b>	Rural net infrared (longwave) radiation	watt/m <sup>2</sup>
<b>FIRA_U</b>	Urban net infrared (longwave) radiation	watt/m <sup>2</sup>
<b>FIRE</b>	emitted infrared (longwave) radiation	watt/m <sup>2</sup>
<b>FIRESEASONL</b>	annual fire season length	days
<b>FIRE_PROB</b>	daily fire probability	0-1
<b>FLDS</b>	atmospheric longwave radiation	watt/m <sup>2</sup>
<b>FLUXFM2A</b>	heat flux for rain to snow conversion	W/m <sup>2</sup>
<b>FLUXFMLND</b>	heat flux from rain to snow conversion	W/m <sup>2</sup>
<b>FNPP</b>	net primary production	g/m <sup>2</sup> /s
<b>FPG</b>	fraction of potential gpp	proportion
<b>FPI</b>	fraction of potential immobilization	proportion
<b>FPSN</b>	photosynthesis	umol/m <sup>2</sup> s
<b>FRAC_ICEOLD</b>	fraction of ice relative to the tot water	proportion
<b>FROOTC</b>	fine root C	gC/m <sup>2</sup>

Name	Long-name	Units
<b>FROOTC</b>	fine root C	gC/m <sup>2</sup>
<b>FROOTC_ALLOC</b>	fine root C allocation	gC/m <sup>2</sup> /s
<b>FROOTC_ALLOC</b>	fine root C allocation	gC/m <sup>2</sup> /s
<b>FROOTC_LOSS</b>	fine root C loss	gC/m <sup>2</sup> /s
<b>FROOTC_LOSS</b>	fine root C loss	gC/m <sup>2</sup> /s
<b>FROOTC_STORAGE</b>	fine root C storage	gC/m <sup>2</sup>
<b>FROOTC_STORAGE_TO_XFER</b>	fine root C shift storage to transfer	gC/m <sup>2</sup> /s
<b>FROOTC_TO_LITR1C</b>	fine root C litterfall to litter 1 C	gC/m <sup>2</sup> /s
<b>FROOTC_TO_LITR2C</b>	fine root C litterfall to litter 2 C	gC/m <sup>2</sup> /s
<b>FROOTC_TO_LITR3C</b>	fine root C litterfall to litter 3 C	gC/m <sup>2</sup> /s
<b>FROOTC_TO_LITTER</b>	fine root C litterfall	gC/m <sup>2</sup> /s
<b>FROOTC_XFER</b>	fine root C transfer	gC/m <sup>2</sup>
<b>FROOTC_XFER_TO_FROOTC</b>	fine root C growth from storage	gC/m <sup>2</sup> /s
<b>FROOTN</b>	fine root N	gN/m <sup>2</sup>
<b>FROOTN_STORAGE</b>	fine root N storage	gN/m <sup>2</sup>
<b>FROOTN_STORAGE_TO_XFER</b>	fine root N shift storage to transfer	gN/m <sup>2</sup> /s
<b>FROOTN_TO_LITR1N</b>	fine root N litterfall to litter 1 N	gN/m <sup>2</sup> /s
<b>FROOTN_TO_LITR2N</b>	fine root N litterfall to litter 2 N	gN/m <sup>2</sup> /s
<b>FROOTN_TO_LITR3N</b>	fine root N litterfall to litter 3 N	gN/m <sup>2</sup> /s
<b>FROOTN_TO_LITTER</b>	fine root N litterfall	gN/m <sup>2</sup> /s
<b>FROOTN_XFER</b>	fine root N transfer	gN/m <sup>2</sup>
<b>FROOTN_XFER_TO_FROOTN</b>	fine root N growth from storage	gN/m <sup>2</sup> /s
<b>FROOT_MR</b>	fine root maintenance respiration	gC/m <sup>2</sup> /s
<b>FSA</b>	absorbed solar radiation	watt/m <sup>2</sup>
<b>FSAT</b>	fractional area with water table at surface	unitless
<b>FSA_R</b>	Rural absorbed solar radiation	watt/m <sup>2</sup>
<b>FSA_U</b>	Urban absorbed solar radiation	watt/m <sup>2</sup>
<b>FSD24</b>	direct radiation (last 24hrs)	K
<b>FSD240</b>	direct radiation (last 240hrs)	K
<b>FSDS</b>	atmospheric incident solar radiation	watt/m <sup>2</sup>
<b>FSDSND</b>	direct nir incident solar radiation	watt/m <sup>2</sup>
<b>FSDSNDLN</b>	direct nir incident solar radiation at local noon	watt/m <sup>2</sup>

Name	Long-name	Units
<b>FSDSNI</b>	diffuse nir incident solar radiation	watt/m <sup>2</sup>
<b>FSDSVD</b>	direct vis incident solar radiation	watt/m <sup>2</sup>
<b>FSDSVDLN</b>	direct vis incident solar radiation at local noon	watt/m <sup>2</sup>
<b>FSDSVI</b>	diffuse vis incident solar radiation	watt/m <sup>2</sup>
<b>FSH</b>	sensible heat	watt/m <sup>2</sup>
<b>FSH_G</b>	sensible heat from ground	watt/m <sup>2</sup>
<b>FSH_NODYNLNDUSE</b>	sensible heat	watt/m <sup>2</sup>
<b>FSH_R</b>	Rural sensible heat	watt/m <sup>2</sup>
<b>FSH_U</b>	Urban sensible heat	watt/m <sup>2</sup>
<b>FSH_V</b>	sensible heat from veg	watt/m <sup>2</sup>
<b>FSI24</b>	indirect radiation (last 24hrs)	K
<b>FSI240</b>	indirect radiation (last 240hrs)	K
<b>FSM</b>	snow melt heat flux	watt/m <sup>2</sup>
<b>FSM_R</b>	Rural snow melt heat flux	watt/m <sup>2</sup>
<b>FSM_U</b>	Urban snow melt heat flux	watt/m <sup>2</sup>
<b>FSNO</b>	fraction of ground covered by snow	unitless
<b>FSR</b>	reflected solar radiation	watt/m <sup>2</sup>
<b>FSRND</b>	direct nir reflected solar radiation	watt/m <sup>2</sup>
<b>FSRNDLN</b>	direct nir reflected solar radiation at local noon	watt/m <sup>2</sup>
<b>FSRNI</b>	diffuse nir reflected solar radiation	watt/m <sup>2</sup>
<b>FSRVD</b>	direct vis reflected solar radiation	watt/m <sup>2</sup>
<b>FSRVDLN</b>	direct vis reflected solar radiation at local noon	watt/m <sup>2</sup>
<b>FSRVI</b>	diffuse vis reflected solar radiation	watt/m <sup>2</sup>
<b>FSUN</b>	sunlit fraction of canopy	proportion
<b>FSUN24</b>	fraction sunlit (last 24hrs)	K
<b>FSUN240</b>	fraction sunlit (last 240hrs)	K
<b>FTDD</b>	down direct flux below veg per unit dir flx	proportion
<b>FTID</b>	down indirect flux below veg per unit dir flx	proportion

Name	Long-name	Units
<b>FTII</b>	down indirect flux below veg per unit indirect flx	proportion
<b>FV</b>	friction velocity for dust model	m/s
<b>FWET</b>	fraction of canopy that is wet	proportion
<b>GAMMA</b>	total gamma for VOC calc	non
<b>GAMMAA</b>	gamma A for VOC calc	non
<b>GAMMAL</b>	gamma L for VOC calc	non
<b>GAMMAP</b>	gamma P for VOC calc	non
<b>GAMMAS</b>	gamma S for VOC calc	non
<b>GAMMAT</b>	gamma T for VOC calc	non
<b>GC_HEAT1</b>	initial gridcell total heat content	J/m <sup>2</sup>
<b>GC_HEAT2</b>	post land cover change total heat content	J/m <sup>2</sup>
<b>GC_ICE1</b>	initial gridcell total ice content	mm
<b>GC_ICE2</b>	post land cover change total ice content	mm
<b>GC_LIQ1</b>	initial gridcell total liq content	mm
<b>GC_LIQ2</b>	initial gridcell total liq content	mm
<b>GDD0</b>	Growing degree days base 0C from planting	ddays
<b>GDD020</b>	Twenty year average of growing degree days base 0C from planting	ddays
<b>GDD10</b>	Growing degree days base 10C from planting	ddays
<b>GDD1020</b>	Twenty year average of growing degree days base 10C from planting	ddays
<b>GDD8</b>	Growing degree days base 8C from planting	ddays
<b>GDD820</b>	Twenty year average of growing degree days base 8C from planting	ddays
<b>GDDHARV</b>	Growing degree days (gdd) needed to harvest	ddays
<b>GDDPLANT</b>	Accumulated growing degree days past planting date for crop	ddays
<b>GDDTSOI</b>	Growing degree-days from planting (top two soil layers)	ddays
<b>GDIR</b>	leaf projection in solar direction	proportion
<b>GPP</b>	gross primary production	gC/m <sup>2</sup> /s

Name	Long-name	Units
<b>GPP</b>	gross primary production	gC/m <sup>2</sup> /s
<b>GR</b>	total growth respiration	gC/m <sup>2</sup> /s
<b>GRESP_STORAGE</b>	growth respiration storage	gC/m <sup>2</sup>
<b>GRESP_STORAGE_TO_XFER</b>	growth respiration shift storage to transfer	gC/m <sup>2</sup> /s
<b>GRESP_XFER</b>	growth respiration transfer	gC/m <sup>2</sup>
<b>GROSS_NMIN</b>	gross rate of N mineralization	gN/m <sup>2</sup> /s
<b>H2OCAN</b>	intercepted water	mm
<b>H2OSNO</b>	snow depth (liquid water)	mm
<b>H2OSNO_TOP</b>	mass of snow in top snow layer	kg/m <sup>2</sup>
<b>H2OSOI</b>	volumetric soil water	mm <sup>3</sup> /mm <sup>3</sup>
<b>HBOT</b>	canopy bottom	m
<b>HC</b>	heat content of soil/snow/lake	MJ/m <sup>2</sup>
<b>HCSOI</b>	soil heat content	MJ/m <sup>2</sup>
<b>HEAT_FROM_AC</b>	sensible heat flux put into canyon due to heat removed from air conditioning	watt/m <sup>2</sup>
<b>HK</b>	hydraulic conductivity	mm/s
<b>HR</b>	total heterotrophic respiration	gC/m <sup>2</sup> /s
<b>HR</b>	total heterotrophic respiration	gC/m <sup>2</sup> /s
<b>HTOP</b>	canopy top	m
<b>INIT_GPP</b>	GPP flux before downregulation	gC/m <sup>2</sup> /s
<b>ISEABEG</b>	index for start of growing season	
<b>ISOPRENE</b>	isoprene flux	uGC/M2/H
<b>LAISHA</b>	shaded projected leaf area index	none
<b>LAISUN</b>	sunlit projected leaf area index	none
<b>LAND_UPTAKE</b>	NEE minus LAND_USE_FLUX, negative for update	gC/m <sup>2</sup> /s
<b>LAND_USE_FLUX</b>	total C emitted from land cover conversion and wood product pools	gC/m <sup>2</sup> /s
<b>LEAFC</b>	leaf C	gC/m <sup>2</sup>
<b>LEAFC</b>	leaf C	gC/m <sup>2</sup>
<b>LEAFC_ALLOC</b>	leaf C allocation	gC/m <sup>2</sup> /s
<b>LEAFC_ALLOC</b>	leaf C allocation	gC/m <sup>2</sup> /s
<b>LEAFC_LOSS</b>	leaf C loss	gC/m <sup>2</sup> /s
<b>LEAFC_LOSS</b>	leaf C loss	gC/m <sup>2</sup> /s
<b>LEAFC_STORAGE</b>	leaf C storage	gC/m <sup>2</sup>
<b>LEAFC_STORAGE_TO_XFER</b>	leaf C shift storage to transfer	gC/m <sup>2</sup> /s

Name	Long-name	Units
<b>LEAFC_TO_LITR1C</b>	leaf C litterfall to litter 1 C	gC/m^2/s
<b>LEAFC_TO_LITR2C</b>	leaf C litterfall to litter 2 C	gC/m^2/s
<b>LEAFC_TO_LITR3C</b>	leaf C litterfall to litter 3 C	gC/m^2/s
<b>LEAFC_TO_LITTER</b>	leaf C litterfall	gC/m^2/s
<b>LEAFC_XFER</b>	leaf C transfer	gC/m^2
<b>LEAFC_XFER_TO_LEAFC</b>	leaf C growth from storage	gC/m^2/s
<b>LEAFN</b>	leaf N	gN/m^2
<b>LEAFN_STORAGE</b>	leaf N storage	gN/m^2
<b>LEAFN_STORAGE_TO_XFER</b>	leaf N shift storage to transfer	gN/m^2/s
<b>LEAFN_TO_LITR1N</b>	leaf N litterfall to litter 1 N	gN/m^2/s
<b>LEAFN_TO_LITR2N</b>	leaf N litterfall to litter 2 N	gN/m^2/s
<b>LEAFN_TO_LITR3N</b>	leaf N litterfall to litter 3 N	gN/m^2/s
<b>LEAFN_TO_LITTER</b>	leaf N litterfall	gN/m^2/s
<b>LEAFN_TO_RETRANSN</b>	leaf N to retranslocated N pool	gN/m^2/s
<b>LEAFN_XFER</b>	leaf N transfer	gN/m^2
<b>LEAFN_XFER_TO_LEAFN</b>	leaf N growth from storage	gN/m^2/s
<b>LEAF_MR</b>	leaf maintenance respiration	gC/m^2/s
<b>LGROW</b>	growing season index (0 or 1)	
<b>LGSF</b>	long growing season factor	proportion
<b>LITFALL</b>	litterfall (leaves and fine roots)	gC/m^2/s
<b>LITFIRE</b>	litter fire losses	gC/m^2/s
<b>LITHR</b>	litter heterotrophic respiration	gC/m^2/s
<b>LITR1C</b>	litter labile C	gC/m^2
<b>LITR1C_TO_SOIL1C</b>	decomp. of litter 1 C to SOM 1 C	gC/m^2/s
<b>LITR1N</b>	litter labile N	gN/m^2
<b>LITR1N_TO_SOIL1N</b>	decomp. of litter 1 N to SOM 1 N	gN/m^2/s
<b>LITR1_HR</b>	het. resp. from litter 1 C	gC/m^2/s
<b>LITR2C</b>	litter cellulose C	gC/m^2
<b>LITR2C_TO_SOIL2C</b>	decomp. of litter 2 C to SOM 2 C	gC/m^2/s
<b>LITR2N</b>	litter cellulose N	gN/m^2
<b>LITR2N_TO_SOIL2N</b>	decomp. of litter 2 N to SOM 2 N	gN/m^2/s
<b>LITR2_HR</b>	het. resp. from litter 2 C	gC/m^2/s
<b>LITR3C</b>	litter lignin C	gC/m^2

Name	Long-name	Units
LITR3C_TO_SOIL3C	decomp. of litter 3 C to SOM 3 C	gC/m^2/s
LITR3N	litter lignin N	gN/m^2
LITR3N_TO_SOIL3N	decomp. of litter 3 N to SOM 3 N	gN/m^2/s
LITR3_HR	het. resp. from litter 3 C	gC/m^2/s
LITTERC	litter C	gC/m^2
LITTERC	litter C	gC/m2
LITTERC_HR	litter C heterotrophic respiration	gC/m^2/s
LITTERC_HR	litter heterotrophic respiration	gC/m2/s
LITTERC_LOSS	litter C loss	gC/m^2/s
LITTERC_LOSS	litter C loss	gC/m2/s
LIVECROOTC	live coarse root C	gC/m^2
LIVECROOTC_STORAGE	live coarse root C storage	gC/m^2
LIVECROOTC_STORAGE_TO_XFER	live coarse root C shift storage to transfer	gC/m^2/s
LIVECROOTC_TO_DEADCROOT	coarse root C turnover	gC/m^2/s
LIVECROOTC_XFER	live coarse root C transfer	gC/m^2
LIVECROOTC_XFER_TO_LIVECROOT	coarse root C growth from storage	gC/m^2/s
LIVECROOTN	live coarse root N	gN/m^2
LIVECROOTN_STORAGE	live coarse root N storage	gN/m^2
LIVECROOTN_STORAGE_TO_XFER	live coarse root N shift storage to transfer	gN/m^2/s
LIVECROOTN_TO_DEADCROOT	coarse root N turnover	gN/m^2/s
LIVECROOTN_TO_RETRANS	live coarse root N to retranslocated N pool	gN/m^2/s
LIVECROOTN_XFER	live coarse root N transfer	gN/m^2
LIVECROOTN_XFER_TO_LIVECROOT	coarse root N growth from storage	gN/m^2/s
LIVECROOT_MR	live coarse root maintenance respiration	gC/m^2/s
LIVEFR	live fraction for leaf, wood, froot	g/m2
LIVESTEMC	live stem C	gC/m^2
LIVESTEMC_STORAGE	live stem C storage	gC/m^2
LIVESTEMC_STORAGE_TO_XFER	stem C shift storage to transfer	gC/m^2/s

Name	Long-name	Units
LIVESTEMC_TO_DEADSTEMC	live stem C turnover	gC/m^2/s
LIVESTEMC_XFER	live stem C transfer	gC/m^2
LIVESTEMC_XFER_TO_LIVESTEMC	live stem C growth from storage	gC/m^2/s
LIVESTEMN	live stem N	gN/m^2
LIVESTEMN_STORAGE	live stem N storage	gN/m^2
LIVESTEMN_STORAGE_TO_XFER	live stem N shift storage to transfer	gN/m^2/s
LIVESTEMN_TO_DEADSTEMN	live stem N turnover	gN/m^2/s
LIVESTEMN_TO_RETRANSN	live stem N to retranslocated N pool	gN/m^2/s
LIVESTEMN_XFER	live stem N transfer	gN/m^2
LIVESTEMN_XFER_TO_LIVESTEMN	live stem N growth from storage	gN/m^2/s
LIVESTEM_MR	live stem maintenance respiration	gC/m^2/s
LNCSHA	leaf N concentration per unit projected LAI	gN/m^2
LNCSUN	leaf N concentration per unit projected LAI	gN/m^2
LWdown	atmospheric longwave radiation	watt/m^2
LWup	upwelling longwave radiation	watt/m^2
ME	moisture of extinction	proportion
MEAN_FIRE_PROB	e-folding mean of daily fire probability	0-1
MONOTERP	monoterpene flux	uGC/M2/H
MR	maintenance respiration	gC/m^2/s
M_CWDC_TO_FIRE	coarse woody debris C fire loss	gC/m^2/s
M_CWDN_TO_FIRE	coarse woody debris N fire loss	gN/m^2/s
M_DEADCROUTC_STORAGE	dead coarse root C storage fire loss	gC/m^2/s
M_DEADCROUTC_STORAGE	dead coarse root C storage mortality to litter 1 C	gC/m^2/s
M_DEADCROUTC_STORAGE	dead coarse root C storage mortality	gC/m^2/s
M_DEADCROUTC_TO_CWDC	dead coarse root C mortality to coarse woody debris C	gC/m^2/s
M_DEADCROUTC_TO_CWDC	dead coarse root C to to woody debris C by fire	gC/m^2/s

Name	Long-name	Units
M_DEADCROOTC_TO_FIRE	dead coarse root C fire loss	gC/m^2/s
M_DEADCROOTC_TO_LITTER	dead coarse root C mortality	gC/m^2/s
M_DEADCROOTC_TO_LITTER	dead coarse root C fire mortality to litter	gC/m^2/s
M_DEADCROOTC_XFER_TO_FIRE	coarse root C transfer fire loss	gC/m^2/s
M_DEADCROOTC_XFER_TO_FIRE	coarse root C transfer mortality to litter 1 C	gC/m^2/s
M_DEADCROOTC_XFER_TO_FIRE	coarse root C transfer mortality	gC/m^2/s
M_DEADCROOTN_STORAGE	root N storage fire loss	gN/m^2/s
M_DEADCROOTN_STORAGE	root N storage mortality to litter 1 N	gN/m^2/s
M_DEADCROOTN_STORAGE	root N storage mortality	gN/m^2/s
M_DEADCROOTN_TO_CWDN	dead coarse root N mortality to coarse woody debris N	gN/m^2/s
M_DEADCROOTN_TO_CWDN	dead coarse root N to woody debris N by fire	gN/m^2/s
M_DEADCROOTN_TO_FIRE	dead coarse root N fire loss	gN/m^2/s
M_DEADCROOTN_TO_LITTER	dead coarse root N mortality	gN/m^2/s
M_DEADCROOTN_TO_LITTER	dead coarse root N fire mortality to litter	gN/m^2/s
M_DEADCROOTN_XFER_TO_FIRE	coarse root N transfer fire loss	gN/m^2/s
M_DEADCROOTN_XFER_TO_FIRE	coarse root N transfer mortality to litter 1 N	gN/m^2/s
M_DEADCROOTN_XFER_TO_FIRE	coarse root N transfer mortality	gN/m^2/s
M_DEADSTEMC_STORAGE	dead stem C storage fire loss	gC/m^2/s
M_DEADSTEMC_STORAGE	dead stem C storage mortality to litter 1 C	gC/m^2/s
M_DEADSTEMC_STORAGE	dead stem C storage mortality	gC/m^2/s
M_DEADSTEMC_TO_CWDC	dead stem C mortality to coarse woody debris C	gC/m^2/s

Name	Long-name	Units
M_DEADSTEMC_TO_CWDC	dead stem C to coarse woody debris C by fire	gC/m^2/s
M_DEADSTEMC_TO_FIRE	dead stem C fire loss	gC/m^2/s
M_DEADSTEMC_TO_LITTER	dead stem C mortality	gC/m^2/s
M_DEADSTEMC_TO_LITTERd	dead stem C fire mortality to litter	gC/m^2/s
M_DEADSTEMC_XFER_TO_FIRE	dead stem C transfer fire loss	gC/m^2/s
M_DEADSTEMC_XFER_TO_LITTER1C	dead stem C transfer mortality to litter 1 C	gC/m^2/s
M_DEADSTEMC_XFER_TO_LITTER	dead stem C transfer mortality	gC/m^2/s
M_DEADSTEMN_STORAGE_TO_FIRE	dead stem N storage fire loss	gN/m^2/s
M_DEADSTEMN_STORAGE_TO_LITTER1N	dead stem N storage mortality to litter 1 N	gN/m^2/s
M_DEADSTEMN_STORAGE_TO_LITTER	dead stem N storage mortality	gN/m^2/s
M_DEADSTEMN_TO_CWDN	dead stem N mortality to coarse woody debris N	gN/m^2/s
M_DEADSTEMN_TO_CWDN	dead stem N to coarse woody debris N by fire	gN/m^2/s
M_DEADSTEMN_TO_FIRE	dead stem N fire loss	gN/m^2/s
M_DEADSTEMN_TO_LITTER	dead stem N mortality	gN/m^2/s
M_DEADSTEMN_TO_LITTERd	dead stem N fire mortality to litter	gN/m^2/s
M_DEADSTEMN_XFER_TO_FIRE	dead stem N transfer fire loss	gN/m^2/s
M_DEADSTEMN_XFER_TO_LITTER1N	dead stem N transfer mortality to litter 1 N	gN/m^2/s
M_DEADSTEMN_XFER_TO_LITTER	dead stem N transfer mortality	gN/m^2/s
M_FROOTC_STORAGE_TO_FIRE	fine root C storage fire loss	gC/m^2/s
M_FROOTC_STORAGE_TO_LITR1C	fine root C storage mortality to litter 1 C	gC/m^2/s
M_FROOTC_STORAGE_TO_LITTER	fine root C storage mortality	gC/m^2/s
M_FROOTC_TO_FIRE	fine root C fire loss	gC/m^2/s
M_FROOTC_TO_LITR1C	fine root C mortality to litter 1 C	gC/m^2/s

Name	Long-name	Units
M_FROOTC_TO_LITR2C	fine root C mortality to litter 2 C	gC/m^2/s
M_FROOTC_TO_LITR3C	fine root C mortality to litter 3 C	gC/m^2/s
M_FROOTC_TO_LITTER	fine root C mortality	gC/m^2/s
M_FROOTC_XFER_TO_FIRE	fine root C transfer fire loss	gC/m^2/s
M_FROOTC_XFER_TO_LITR1C	fine root C transfer mortality to litter 1 C	gC/m^2/s
M_FROOTC_XFER_TO_LITTER	fine root C transfer mortality	gC/m^2/s
M_FROOTN_STORAGE_TO_FIRE	fine root N storage fire loss	gN/m^2/s
M_FROOTN_STORAGE_TO_LITR1N	fine root N storage mortality to litter 1 N	gN/m^2/s
M_FROOTN_STORAGE_TO_LITTER	fine root N storage mortality	gN/m^2/s
M_FROOTN_TO_FIRE	fine root N fire loss	gN/m^2/s
M_FROOTN_TO_LITR1N	fine root N mortality to litter 1 N	gN/m^2/s
M_FROOTN_TO_LITR2N	fine root N mortality to litter 2 N	gN/m^2/s
M_FROOTN_TO_LITR3N	fine root N mortality to litter 3 N	gN/m^2/s
M_FROOTN_TO_LITTER	fine root N mortality	gN/m^2/s
M_FROOTN_XFER_TO_FIRE	fine root N transfer fire loss	gN/m^2/s
M_FROOTN_XFER_TO_LITR1N	fine root N transfer mortality to litter 1 N	gN/m^2/s
M_FROOTN_XFER_TO_LITTER	fine root N transfer mortality	gN/m^2/s
M_GRESP_STORAGE_TO_FIRE	growth respiration storage fire loss	gC/m^2/s
M_GRESP_STORAGE_TO_LITR1C	growth respiration storage mortality to litter 1 C	gC/m^2/s
M_GRESP_STORAGE_TO_LITTER	growth respiration storage mortality	gC/m^2/s
M_GRESP_XFER_TO_FIRE	growth respiration transfer fire loss	gC/m^2/s
M_GRESP_XFER_TO_LITR1C	growth respiration transfer mortality to litter 1 C	gC/m^2/s
M_GRESP_XFER_TO_LITTER	growth respiration transfer mortality	gC/m^2/s
M_LEAFC_STORAGE_TO_FIRE	leaf C storage fire loss	gC/m^2/s

Name	Long-name	Units
M_LEAFC_STORAGE_TO_LITTERC	leaf C storage mortality to litter 1 C	gC/m^2/s
M_LEAFC_STORAGE_TO_LITTER	leaf C storage mortality	gC/m^2/s
M_LEAFC_TO_FIRE	leaf C fire loss	gC/m^2/s
M_LEAFC_TO_LITR1C	leaf C mortality to litter 1 C	gC/m^2/s
M_LEAFC_TO_LITR2C	leaf C mortality to litter 2 C	gC/m^2/s
M_LEAFC_TO_LITR3C	leaf C mortality to litter 3 C	gC/m^2/s
M_LEAFC_TO_LITTER	leaf C mortality	gC/m^2/s
M_XFER_TO_FIRE	leaf C transfer fire loss	gC/m^2/s
M_XFER_TO_LITR1C	leaf C transfer mortality to litter 1 C	gC/m^2/s
M_XFER_TO_LITTER	leaf C transfer mortality	gC/m^2/s
M_LEAFN_STORAGE_TO_FIRE	leaf N storage fire loss	gN/m^2/s
M_LEAFN_STORAGE_TO_LITTERN	leaf N storage mortality to litter 1 N	gN/m^2/s
M_LEAFN_STORAGE_TO_LITTER	leaf N storage mortality	gN/m^2/s
M_LEAFN_TO_FIRE	leaf N fire loss	gN/m^2/s
M_LEAFN_TO_LITR1N	leaf N mortality to litter 1 N	gN/m^2/s
M_LEAFN_TO_LITR2N	leaf N mortality to litter 2 N	gN/m^2/s
M_LEAFN_TO_LITR3N	leaf N mortality to litter 3 N	gN/m^2/s
M_LEAFN_TO_LITTER	leaf N mortality	gN/m^2/s
M_XFER_TO_FIRE	leaf N transfer fire loss	gN/m^2/s
M_XFER_TO_LITR1N	leaf N transfer mortality to litter 1 N	gN/m^2/s
M_XFER_TO_LITTER	leaf N transfer mortality	gN/m^2/s
M_LITR1C_TO_FIRE	litter 1 C fire loss	gC/m^2/s
M_LITR1N_TO_FIRE	litter 1 N fire loss	gN/m^2/s
M_LITR2C_TO_FIRE	litter 2 C fire loss	gC/m^2/s
M_LITR2N_TO_FIRE	litter 2 N fire loss	gN/m^2/s
M_LITR3C_TO_FIRE	litter 3 C fire loss	gC/m^2/s
M_LITR3N_TO_FIRE	litter 3 N fire loss	gN/m^2/s
M_LIVECROOTC_STORAGE	root C storage fire loss	gC/m^2/s
M_LIVECROOTC_STORAGE	root C storage mortality to litter 1 C	gC/m^2/s

Name	Long-name	Units
M_LIVECROOTC_STORAGE	<del>TO_dolTER</del> C storage mortality	gC/m^2/s
M_LIVECROOTC_TO_CWDC	live coarse root C mortality to coarse woody debris C	gC/m^2/s
M_LIVECROOTC_TO_FIRE	live coarse root C fire loss	gC/m^2/s
M_LIVECROOTC_TO_LITTER	live coarse root C mortality	gC/m^2/s
M_LIVECROOTC_XFER_TO	<del>FIRE</del> coarse root C transfer fire loss	gC/m^2/s
M_LIVECROOTC_XFER_TO	<del>LITTER</del> coarse root C transfer mortality to litter 1 C	gC/m^2/s
M_LIVECROOTC_XFER_TO	<del>LITTER</del> coarse root C transfer mortality	gC/m^2/s
M_LIVECROOTN_STORAGE	<del>TO_FIRE</del> root N storage fire loss	gN/m^2/s
M_LIVECROOTN_STORAGE	<del>TO_dolTER</del> root N storage mortality to litter 1 N	gN/m^2/s
M_LIVECROOTN_STORAGE	<del>TO_dolTER</del> root N storage mortality	gN/m^2/s
M_LIVECROOTN_TO_CWDN	live coarse root N mortality to coarse woody debris N	gN/m^2/s
M_LIVECROOTN_TO_FIRE	live coarse root N fire loss	gN/m^2/s
M_LIVECROOTN_TO_LITTER	live coarse root N mortality	gN/m^2/s
M_LIVECROOTN_XFER_TO	<del>FIRE</del> coarse root N transfer fire loss	gN/m^2/s
M_LIVECROOTN_XFER_TO	<del>LITTER</del> coarse root N transfer mortality to litter 1 N	gN/m^2/s
M_LIVECROOTN_XFER_TO	<del>LITTER</del> coarse root N transfer mortality	gN/m^2/s
M_LIVESTEMC_STORAGE_TO_FIRE	C storage fire loss	gC/m^2/s
M_LIVESTEMC_STORAGE_TO_LITTER	<del>TO_dolTER</del> C storage mortality to litter 1 C	gC/m^2/s
M_LIVESTEMC_STORAGE_TO_LITTER	<del>TO_dolTER</del> storage mortality	gC/m^2/s
M_LIVESTEMC_TO_CWDC	live stem C mortality to coarse woody debris C	gC/m^2/s
M_LIVESTEMC_TO_FIRE	live stem C fire loss	gC/m^2/s
M_LIVESTEMC_TO_LITTER	live stem C mortality	gC/m^2/s

Name	Long-name	Units
M_LIVESTEMC_XFER_TO_FIRE	stem C transfer fire loss	gC/m^2/s
M_LIVESTEMC_XFER_TO_LITTER1C	Cm C transfer mortality to litter 1 C	gC/m^2/s
M_LIVESTEMC_XFER_TO_LITTER2C	m C transfer mortality	gC/m^2/s
M_LIVESTEMN_STORAGE_TO_FIRE	N storage fire loss	gN/m^2/s
M_LIVESTEMN_STORAGE_TO_LITTER1N	storage mortality to litter 1 N	gN/m^2/s
M_LIVESTEMN_STORAGE_TO_LITTER2N	storage mortality	gN/m^2/s
M_LIVESTEMN_TO_CWDN	live stem N mortality to coarse woody debris N	gN/m^2/s
M_LIVESTEMN_TO_FIRE	live stem N fire loss	gN/m^2/s
M_LIVESTEMN_TO_LITTER	live stem N mortality	gN/m^2/s
M_LIVESTEMN_XFER_TO_FIRE	stem N transfer fire loss	gN/m^2/s
M_LIVESTEMN_XFER_TO_LITTER1N	Nm N transfer mortality to litter 1 N	gN/m^2/s
M_LIVESTEMN_XFER_TO_LITTER2N	Nm N transfer mortality	gN/m^2/s
M_RETRANSN_TO_FIRE	retranslocated N pool fire loss	gN/m^2/s
M_RETRANSN_TO_LITR1N	retranslocated N pool mortality to litter 1 N	gN/m^2/s
M_RETRANSN_TO_LITTER	retranslocated N pool mortality	gN/m^2/s
NBP	net biome production, includes fire, landuse, and harvest flux, positive for sink	gC/m^2/s
NDEPLOY	total N deployed in new growth	gN/m^2/s
NDEP_TO_SMINN	atmospheric N deposition to soil mineral N	gN/m^2/s
NEE	net ecosystem exchange of carbon, includes fire, landuse, harvest, and hrv_xsmpool flux, positive for source	gC/m^2/s
NEE	net ecosystem exchange (GPP, Re, and fire)	gC/m2/s
NEP	net ecosystem production, excludes fire, landuse, and harvest flux, positive for sink	gC/m^2/s

Name	Long-name	Units
NEP	net ecosystem production	gC/m <sup>2</sup> /s
NET_NMIN	net rate of N mineralization	gN/m <sup>2</sup> /s
NFIX_TO_SMINN	symbiotic/asymbiotic N fixation to soil mineral N	gN/m <sup>2</sup> /s
NPOOL	temporary plant N pool	gN/m <sup>2</sup>
NPOOL_TO_DEADCROOTN	allocation to dead coarse root N	gN/m <sup>2</sup> /s
NPOOL_TO_DEADCROOTN_STORAGE	allocation to dead coarse root N storage	gN/m <sup>2</sup> /s
NPOOL_TO_DEADSTEMN	allocation to dead stem N	gN/m <sup>2</sup> /s
NPOOL_TO_DEADSTEMN_STORAGE	allocation to dead stem N storage	gN/m <sup>2</sup> /s
NPOOL_TO_FROOTN	allocation to fine root N	gN/m <sup>2</sup> /s
NPOOL_TO_FROOTN_STORAGE	allocation to fine root N storage	gN/m <sup>2</sup> /s
NPOOL_TO_LEAFN	allocation to leaf N	gN/m <sup>2</sup> /s
NPOOL_TO_LEAFN_STORAGE	allocation to leaf N storage	gN/m <sup>2</sup> /s
NPOOL_TO_LIVECROOTN	allocation to live coarse root N	gN/m <sup>2</sup> /s
NPOOL_TO_LIVECROOTN_STORAGE	allocation to live coarse root N storage	gN/m <sup>2</sup> /s
NPOOL_TO_LIVESTEMN	allocation to live stem N	gN/m <sup>2</sup> /s
NPOOL_TO_LIVESTEMN_STORAGE	allocation to live stem N storage	gN/m <sup>2</sup> /s
NPP	net primary production	gC/m <sup>2</sup> /s
NPP	net primary production	gC/m <sup>2</sup> /s
NSTEPBEG	nstep at start of growing season	
N_ALLOMETRY	N allocation index	none
OCDEP	total OC deposition (dry+wet) from atmosphere	kg/m <sup>2</sup> /s
OFFSET_COUNTER	offset days counter	days
OFFSET_FDD	offset freezing degree days counter	C degree-days
OFFSET_FLAG	offset flag	none
OFFSET_SWI	offset soil water index	none
OMEGA	fraction of intercepted radiation that is scattered	proportion
ONSET_COUNTER	onset days counter	days
ONSET_FDD	onset freezing degree days counter	C degree-days
ONSET_FLAG	onset flag	none

Name	Long-name	Units
<b>ONSET_GDD</b>	onset growing degree days	C degree-days
<b>ONSET_GDDFLAG</b>	onset flag for growing degree day sum	none
<b>ONSET_SWI</b>	onset soil water index	none
<b>ORVOC</b>	other reactive VOC flux	uGC/M2/H
<b>OVOC</b>	other VOC flux	uGC/M2/H
<b>PAR240_shade</b>	shade PAR (240 hrs)	umol/m2/s
<b>PAR240_sun</b>	sunlit PAR (240 hrs)	umol/m2/s
<b>PAR24_shade</b>	shade PAR (24 hrs)	umol/m2/s
<b>PAR24_sun</b>	sunlit PAR (24 hrs)	umol/m2/s
<b>PARSHA</b>	average absorbed PAR for shaded leaves	W/m^2
<b>PARSUN</b>	average absorbed PAR for sunlit leaves	W/m^2
<b>PAR_shade</b>	shade PAR	umol/m2/s
<b>PAR_sun</b>	sunlit PAR	umol/m2/s
<b>PBOT</b>	atmospheric pressure	Pa
<b>PCO2</b>	atmospheric partial pressure of CO2	Pa
<b>PET</b>	Potential Evapotranspiration	mm h2o/s
<b>PFT_CTRUNC</b>	pft-level sink for C truncation	gC/m^2
<b>PFT_FIRE_CLOSS</b>	total pft-level fire C loss	gC/m^2/s
<b>PFT_FIRE_NLOSS</b>	total pft-level fire N loss	gN/m^2/s
<b>PFT_NTRUNC</b>	pft-level sink for N truncation	gN/m^2
<b>PLAI</b>	Prognostic Leaf Area Index	m2/m2
<b>PLANT_CALLOC</b>	total allocated C flux	gC/m^2/s
<b>PLANT_NALLOC</b>	total allocated N flux	gN/m^2/s
<b>PLANT_NDEMAND</b>	N flux required to support initial GPP	gN/m^2/s
<b>POTENTIAL_IMMOB</b>	potential N immobilization	gN/m^2/s
<b>PREV_DAYL</b>	daylength from previous timestep	s
<b>PREV_FROOTC_TO_LITTER</b>	previous timestep froot C litterfall flux	gC/m^2/s
<b>PREV_LEAFC_TO_LITTER</b>	previous timestep leaf C litterfall flux	gC/m^2/s
<b>PROD100C</b>	100-yr wood product C	gC/m^2
<b>PROD100C_LOSS</b>	loss from 100-yr wood product pool	gC/m^2/s
<b>PROD100N</b>	100-yr wood product N	gN/m^2

Name	Long-name	Units
<b>PROD100N_LOSS</b>	loss from 100-yr wood product pool	gN/m^2/s
<b>PROD10C</b>	10-yr wood product C	gC/m^2
<b>PROD10C LOSS</b>	loss from 10-yr wood product pool	gC/m^2/s
<b>PROD10N</b>	10-yr wood product N	gN/m^2
<b>PROD10N LOSS</b>	loss from 10-yr wood product pool	gN/m^2/s
<b>PRODUCT_CLOSS</b>	total carbon loss from wood product pools	gC/m^2/s
<b>PRODUCT_NLOSS</b>	total N loss from wood product pools	gN/m^2/s
<b>PSNSHA</b>	shaded leaf photosynthesis	umolCO2/m^2/s
<b>PSNSHADE_TO_CPOOL</b>	C fixation from shaded canopy	gC/m^2/s
<b>PSNSUN</b>	sunlit leaf photosynthesis	umolCO2/m^2/s
<b>PSNSUN_TO_CPOOL</b>	C fixation from sunlit canopy	gC/m^2/s
<b>PSurf</b>	surface pressure	Pa
<b>Q2M</b>	2m specific humidity	kg/kg
<b>QBOT</b>	atmospheric specific humidity	kg/kg
<b>QCHANR</b>	RTM river flow: LIQ	m3/s
<b>QCHANR_ICE</b>	RTM river flow: ICE	m3/s
<b>QCHARGE</b>	aquifer recharge rate	mm/s
<b>QCHOCNR</b>	RTM river discharge into ocean: LIQ	m3/s
<b>QCHOCNR_ICE</b>	RTM river discharge into ocean: ICE	m3/s
<b>QDRAI</b>	sub-surface drainage	mm/s
<b>QDRIP</b>	throughfall	mm/s
<b>QFLX_DEW_GRND</b>	ground surface dew formation	mm H2O/s
<b>QFLX_DEW_SNOW</b>	surface dew added to snow pacK	mm H2O/s
<b>QFLX_EVAP_TOT</b>	qflx_evap_soi + qflx_evap_can + qflx_tran_veg	mm H2O/s
<b>QFLX_EVAP_VEG</b>	vegetation evaporation	mm H2O/s
<b>QFLX_ICE_DYNBAL</b>	ice dynamic land cover change conversion runoff flux	mm/s
<b>QFLX_LIQ_DYNBAL</b>	liq dynamic land cover change conversion runoff flux	mm/s
<b>QFLX_RAIN_GRND</b>	rain on ground after interception	mm H2O/s
<b>QFLX_SNOW_GRND</b>	snow on ground after interception	mm H2O/s

Name	Long-name	Units
<b>QFLX_SUB_SNOW</b>	sublimation rate from snow pack	mm H <sub>2</sub> O/s
<b>QICE</b>	ice growth/melt	mm/s
<b>QICEYR</b>	ice growth/melt	mm/s
<b>QINFL</b>	infiltration	mm/s
<b>QINTR</b>	interception	mm/s
<b>QIRRIG</b>	water added through irrigation	mm/s
<b>QMELT</b>	snow melt	mm/s
<b>QOVER</b>	surface runoff	mm/s
<b>QRGWL</b>	surface runoff at glaciers (liquid only), wetlands, lakes	mm/s
<b>QRUNOFF</b>	total liquid runoff (does not include QSNWCPICE)	mm/s
<b>QRUNOFF_NODYNLNDUSE</b>	total liquid runoff (does not include QSNWCPICE)	mm/s
<b>QRUNOFF_R</b>	Rural total runoff	mm/s
<b>QRUNOFF_U</b>	Urban total runoff	mm/s
<b>QSNWCPICE</b>	excess snowfall due to snow capping	mm/s
<b>QSNWCPICE_NODYNLNDUSE</b>	Excess snowfall due to snow capping	mm H <sub>2</sub> O/s
<b>QSNWCPLIQ</b>	excess rainfall due to snow capping	mm H <sub>2</sub> O/s
<b>QSOIL</b>	ground evaporation	mm/s
<b>QTOPSOIL</b>	water input to surface	mm/s
<b>QVEGE</b>	canopy evaporation	mm/s
<b>QVEGT</b>	canopy transpiration	mm/s
<b>Qair</b>	atmospheric specific humidity	kg/kg
<b>Qanth</b>	anthropogenic heat flux	watt/m <sup>2</sup>
<b>Qh</b>	sensible heat	watt/m <sup>2</sup>
<b>Qle</b>	total evaporation	watt/m <sup>2</sup>
<b>Qstor</b>	storage heat flux (includes snowmelt)	watt/m <sup>2</sup>
<b>Qtau</b>	momentum flux	kg/m/s <sup>2</sup>
<b>RAIN</b>	atmospheric rain	mm/s
<b>RAINATM</b>	atmospheric rain forcing	mm/s
<b>RAINATM</b>	atmospheric rain forcing	mm/s
<b>RAINFM2A</b>	land rain on atm grid	mm/s
<b>RAM1</b>	aerodynamical resistance	s/m
<b>RC13_CANAIR</b>	C13/C(12+13) for canopy air	proportion

Name	Long-name	Units
<b>RC13_PSNSHA</b>	C13/C(12+13) for shaded photosynthesis	proportion
<b>RC13_PSNSUN</b>	C13/C(12+13) for sunlit photosynthesis	proportion
<b>RESP_C</b>	Amt. of Carbon lost to atm by pool	g/m <sup>2</sup> /s
<b>RETRANSN</b>	plant pool of retranslocated N	gN/m <sup>2</sup>
<b>RETRANSN_TO_NPOOL</b>	deployment of retranslocated N	gN/m <sup>2</sup> /s
<b>RH</b>	atmospheric relative humidity	%
<b>RH2M</b>	2m relative humidity	%
<b>RH2M_R</b>	Rural 2m specific humidity	%
<b>RH2M_U</b>	Urban 2m relative humidity	%
<b>ROOTFR</b>	fraction of roots in each soil layer	proportion
<b>ROOTR</b>	effective fraction of roots in each soil layer	proportion
<b>ROOTR_COLUMN</b>	effective fraction of roots in each soil layer	proportion
<b>RR</b>	root respiration (fine root MR + total root GR)	gC/m <sup>2</sup> /s
<b>RRESIS</b>	root resistance in each soil layer	proportion
<b>RSSHA</b>	shaded leaf stomatal resistance	s/m
<b>RSSUN</b>	sunlit leaf stomatal resistance	s/m
<b>Rainf</b>	atmospheric rain	mm/s
<b>Rnet</b>	net radiation	watt/m <sup>2</sup>
<b>SABG</b>	solar rad absorbed by ground	watt/m <sup>2</sup>
<b>SABV</b>	solar rad absorbed by veg	watt/m <sup>2</sup>
<b>SEEDC</b>	pool for seeding new PFTs	gC/m <sup>2</sup>
<b>SEEDN</b>	pool for seeding new PFTs	gN/m <sup>2</sup>
<b>SHA_AID</b>	shade canopy absorbed indirect from direct	W/m <sup>2</sup>
<b>SHA_AII</b>	shade canopy absorbed indirect from indirect	W/m <sup>2</sup>
<b>SHA_ALF</b>	shade canopy total absorbed by leaves	W/m <sup>2</sup>
<b>SHA_APERLAI</b>	shade canopy total absorbed per unit LAI	W/m <sup>2</sup>
<b>SHA_ATOT</b>	shade canopy total absorbed	W/m <sup>2</sup>
<b>SHA_FAID</b>	fraction shade canopy absorbed indirect from direct	proportion

Name	Long-name	Units
<b>SHA_FAII</b>	fraction shade canopy absorbed indirect from indirect	proportion
<b>SLASHA</b>	specific leaf area for shaded canopy, projected area basis	$m^2/gC$
<b>SLASUN</b>	specific leaf area for sunlit canopy, projected area basis	$m^2/gC$
<b>SMINN</b>	soil mineral N	$gN/m^2$
<b>SMINN_LEACHED</b>	soil mineral N pool loss to leaching	$gN/m^2/s$
<b>SMINN_TO_DENIT_EXCESS</b>	denitrification from excess mineral N pool	$gN/m^2/s$
<b>SMINN_TO_DENIT_L1S1</b>	denitrification for decomp. of litter 1 to SOM 1	$gN/m^2/s$
<b>SMINN_TO_DENIT_L2S2</b>	denitrification for decomp. of litter 2 to SOM 2	$gN/m^2/s$
<b>SMINN_TO_DENIT_L3S3</b>	denitrification for decomp. of litter 3 to SOM 3	$gN/m^2/s$
<b>SMINN_TO_DENIT_S1S2</b>	denitrification for decomp. of SOM 1 to SOM 2	$gN/m^2/s$
<b>SMINN_TO_DENIT_S2S3</b>	denitrification for decomp. of SOM 2 to SOM 3	$gN/m^2/s$
<b>SMINN_TO_DENIT_S3S4</b>	denitrification for decomp. of SOM 3 to SOM 4	$gN/m^2/s$
<b>SMINN_TO_DENIT_S4</b>	denitrification for decomp. of SOM 4	$gN/m^2/s$
<b>SMINN_TO_NPOOL</b>	deployment of soil mineral N uptake	$gN/m^2/s$
<b>SMINN_TO_PLANT</b>	plant uptake of soil mineral N	$gN/m^2/s$
<b>SMINN_TO_SOIL1N_L1</b>	mineral N flux for decomp. of litter 1 to SOM 1	$gN/m^2/s$
<b>SMINN_TO_SOIL2N_L2</b>	mineral N flux for decomp. of litter 2 to SOM 2	$gN/m^2/s$
<b>SMINN_TO_SOIL2N_S1</b>	mineral N flux for decomp. of SOM 1 to SOM 2	$gN/m^2/s$
<b>SMINN_TO_SOIL3N_L3</b>	mineral N flux for decomp. of litter 3 to SOM 3	$gN/m^2/s$
<b>SMINN_TO_SOIL3N_S2</b>	mineral N flux for decomp. of SOM 2 to SOM 3	$gN/m^2/s$
<b>SMINN_TO_SOIL4N_S3</b>	mineral N flux for decomp. of SOM 3 to SOM 4	$gN/m^2/s$
<b>SMOIST</b>	Soil moisture for top 30cm	$mm^3/mm^3$
<b>SMP</b>	soil matric potential	mm

Name	Long-name	Units
<b>SNOAERFRC2L</b>	surface forcing of all aerosols in snow, averaged only when snow is present (land)	watt/m <sup>2</sup>
<b>SNOAERFRCL</b>	surface forcing of all aerosols in snow (land)	watt/m <sup>2</sup>
<b>SNOBCFRC2L</b>	surface forcing of BC in snow, averaged only when snow is present (land)	watt/m <sup>2</sup>
<b>SNOBCFRCL</b>	surface forcing of BC in snow (land)	watt/m <sup>2</sup>
<b>SNOBCMCL</b>	mass of BC in snow column	kg/m <sup>2</sup>
<b>SNOBCMSL</b>	mass of BC in top snow layer	kg/m <sup>2</sup>
<b>SNODSTFRC2L</b>	surface forcing of dust in snow, averaged only when snow is present (land)	watt/m <sup>2</sup>
<b>SNODSTFRCL</b>	surface forcing of dust in snow (land)	watt/m <sup>2</sup>
<b>SNODSTMCL</b>	mass of dust in snow column	kg/m <sup>2</sup>
<b>SNODSTMSL</b>	mass of dust in top snow layer	kg/m <sup>2</sup>
<b>SNOFSDSND</b>	direct nir incident solar radiation on snow	watt/m <sup>2</sup>
<b>SNOFSDSNI</b>	diffuse nir incident solar radiation on snow	watt/m <sup>2</sup>
<b>SNOFSDSVD</b>	direct vis incident solar radiation on snow	watt/m <sup>2</sup>
<b>SNOFSDSVI</b>	diffuse vis incident solar radiation on snow	watt/m <sup>2</sup>
<b>SNOFSRND</b>	direct nir reflected solar radiation from snow	watt/m <sup>2</sup>
<b>SNOFSRNI</b>	diffuse nir reflected solar radiation from snow	watt/m <sup>2</sup>
<b>SNOFSRVD</b>	direct vis reflected solar radiation from snow	watt/m <sup>2</sup>
<b>SNOFSRVI</b>	diffuse vis reflected solar radiation from snow	watt/m <sup>2</sup>
<b>SNOLIQFL</b>	top snow layer liquid water fraction (land)	fraction
<b>SNOOCFRC2L</b>	surface forcing of OC in snow, averaged only when snow is present (land)	watt/m <sup>2</sup>
<b>SNOOCFRCL</b>	surface forcing of OC in snow (land)	watt/m <sup>2</sup>

Name	Long-name	Units
<b>SNOOCMCL</b>	mass of OC in snow column	kg/m2
<b>SNOOCMSL</b>	mass of OC in top snow layer	kg/m2
<b>SNORDSL</b>	top snow layer effective grain radius	m^-6
<b>SNOTTOPL</b>	snow temperature (top layer)	K/m
<b>SNOW</b>	atmospheric snow	mm/s
<b>SNOWATM</b>	atmospheric snow forcing	mm/s
<b>SNOWATM</b>	atmospheric snow forcing	mm/s
<b>SNOWDP</b>	snow height	m
<b>SNOWFM2A</b>	land snow on atm grid	mm/s
<b>SNOWICE</b>	snow ice	kg/m2
<b>SNOWLIQ</b>	snow liquid water	kg/m2
<b>SNOdTdzL</b>	top snow layer temperature gradient (land)	K/m
<b>SOIL1C</b>	soil organic matter C (fast pool)	gC/m^2
<b>SOIL1C_TO_SOIL2C</b>	decomp. of SOM 1 C to SOM 2 C	gC/m^2/s
<b>SOIL1N</b>	soil organic matter N (fast pool)	gN/m^2
<b>SOIL1N_TO_SOIL2n</b>	decomp. of SOM 1 N to SOM 2 N	gN/m^2/s
<b>SOIL1_HR</b>	het. resp. from SOM 1 C	gC/m^2/s
<b>SOIL2C</b>	soil organic matter C (medium pool)	gC/m^2
<b>SOIL2C_TO_SOIL3C</b>	decomp. of SOM 2 C to SOM 3 C	gC/m^2/s
<b>SOIL2N</b>	soil organic matter N (medium pool)	gN/m^2
<b>SOIL2N_TO_SOIL3N</b>	decomp. of SOM 2 N to SOM 3 N	gN/m^2/s
<b>SOIL2_HR</b>	het. resp. from SOM 2 C	gC/m^2/s
<b>SOIL3C</b>	soil organic matter C (slow pool)	gC/m^2
<b>SOIL3C_TO_SOIL4C</b>	decomp. of SOM 3 C to SOM 4 C	gC/m^2/s
<b>SOIL3N</b>	soil orgainc matter N (slow pool)	gN/m^2
<b>SOIL3N_TO_SOIL4N</b>	decomp. of SOM 3 N to SOM 4 N	gN/m^2/s
<b>SOIL3_HR</b>	het. resp. from SOM 3 C	gC/m^2/s
<b>SOIL4C</b>	soil organic matter C (slowest pool)	gC/m^2

Name	Long-name	Units
<b>SOIL4N</b>	soil organic matter N (slowest pool)	gN/m <sup>2</sup>
<b>SOIL4N_TO_SMINN</b>	N mineralization for decomp. of SOM 4	gN/m <sup>2</sup> /s
<b>SOIL4_HR</b>	het. resp. from SOM 4 C	gC/m <sup>2</sup> /s
<b>SOILC</b>	soil C	gC/m <sup>2</sup>
<b>SOILC</b>	total soil organic matter C (excluding CWDC and LITTERC)	gC/m <sup>2</sup>
<b>SOILC_HR</b>	soil C heterotrophic respiration	gC/m <sup>2</sup> /s
<b>SOILC_HR</b>	soil heterotrophic respiration	gC/m <sup>2</sup> /s
<b>SOILC_LOSS</b>	soil C loss	gC/m <sup>2</sup> /s
<b>SOILC_LOSS</b>	soil C loss	gC/m <sup>2</sup> /s
<b>SOILICE</b>	soil ice	kg/m <sup>2</sup>
<b>SOILLIQ</b>	soil liquid water	kg/m <sup>2</sup>
<b>SOILPSI</b>	soil water potential in each soil layer	MPa
<b>SOILT</b>	Soil temperature for top 30cm	deg C
<b>SOILWATER_10CM</b>	soil liquid water + ice in top 10cm of soil	kg/m <sup>2</sup>
<b>SOMFIRE</b>	soil organic matter fire losses	gC/m <sup>2</sup> /s
<b>SOMHR</b>	soil organic matter heterotrophic respiration	gC/m <sup>2</sup> /s
<b>SR</b>	total soil respiration (HR + root resp)	gC/m <sup>2</sup> /s
<b>STORAGE_GR</b>	growth resp for growth sent to storage for later display	gC/m <sup>2</sup> /s
<b>STORVEGC</b>	stored vegetation carbon, excluding cpool	gC/m <sup>2</sup>
<b>STORVEGN</b>	stored vegetation nitrogen	gN/m <sup>2</sup>
<b>STRESSCD</b>	cold + drought stress function for leaf loss	
<b>STRESST</b>	temperature stress function for leaf loss	
<b>STRESSW</b>	water stress function for leaf loss	
<b>SUN_ADD</b>	sun canopy absorbed direct from direct	W/m <sup>2</sup>
<b>SUN_AID</b>	sun canopy absorbed indirect from direct	W/m <sup>2</sup>
<b>SUN_AII</b>	sun canopy absorbed indirect from indirect	W/m <sup>2</sup>

Name	Long-name	Units
SUN_ALF	sun canopy total absorbed by leaves	W/m^2
SUN_APERLAI	sun canopy total absorbed per unit LAI	W/m^2
SUN_ATOT	sun canopy total absorbed	W/m^2
SUN_FAID	fraction sun canopy absorbed indirect from direct	proportion
SUN_FAII	fraction sun canopy absorbed indirect from indirect	proportion
SUPPLEMENT_TO_SMINN	supplemental N supply	gN/m^2/s
SWdown	atmospheric incident solar radiation	watt/m^2
SWup	upwelling shortwave radiation	watt/m^2
SoilAlpha	factor limiting ground evap	unitless
SoilAlpha_U	urban factor limiting ground evap	unitless
T10	10-day running mean of 2-m temperature	K
TAUX	zonal surface stress	kg/m/s^2
TAUY	meridional surface stress	kg/m/s^2
TBOT	atmospheric air temperature	K
TBUILD	internal urban building temperature	K
TDA	daily average 2-m temperature	K
TDAYAVG	Daily Averaged Temperature	deg C
TEMPAVG_T2M	temporary average 2m air temperature	K
TEMPMAX_RETRANSN	temporary annual max of retranslocated N pool	gN/m^2
TEMPSUM_POTENTIAL_GPP	temporary annual sum of potential GPP	gC/m^2/yr
TG	ground temperature	K
TG_R	Rural ground temperature	K
TG_U	Urban ground temperature	K
THBOT	atmospheric air potential temperature	K
TLAI	total projected leaf area index	none
TLAKE	lake temperature	K
TOPT	topt coefficient for VOC calc	non
TOTCOLC	total column carbon, incl veg and cpool	gC/m^2

Name	Long-name	Units
<b>TOTCOLN</b>	total column-level N	gN/m <sup>2</sup>
<b>TOTECOSYSC</b>	total ecosystem carbon, incl veg but excl cpool	gC/m <sup>2</sup>
<b>TOTECOSYSN</b>	total ecosystem N	gN/m <sup>2</sup>
<b>TOTFIRE</b>	total ecosystem fire losses	gC/m <sup>2</sup> /s
<b>TOTLITC</b>	total litter carbon	gC/m <sup>2</sup>
<b>TOTLITN</b>	total litter N	gN/m <sup>2</sup>
<b>TOTPFTC</b>	total pft-level carbon, including cpool	gC/m <sup>2</sup>
<b>TOTPFTN</b>	total PFT-level nitrogen	gN/m <sup>2</sup>
<b>TOTPDOC</b>	total wood product C	gC/m <sup>2</sup>
<b>TOTPORDN</b>	total wood product N	gN/m <sup>2</sup>
<b>TOTSOMC</b>	total soil organic matter carbon	gC/m <sup>2</sup>
<b>TOTSOMN</b>	total soil organic matter N	gN/m <sup>2</sup>
<b>TOTVEGC</b>	total vegetation carbon, excluding cpool	gC/m <sup>2</sup>
<b>TOTVEGN</b>	total vegetation nitrogen	gN/m <sup>2</sup>
<b>TOT_AID</b>	total canopy absorbed indirect from direct	W/m <sup>2</sup>
<b>TPOOL_C</b>	Total Carbon for pool	g/m <sup>2</sup>
<b>TRAFFICFLUX</b>	sensible heat flux from urban traffic	watt/m <sup>2</sup>
<b>TRANSFER_DEADCROOT_GR</b>	dead coarse root growth respiration from storage	gC/m <sup>2</sup> /s
<b>TRANSFER_DEADSTEM_GR</b>	dead stem growth respiration from storage	gC/m <sup>2</sup> /s
<b>TRANSFER_FROOT_GR</b>	fine root growth respiration from storage	gC/m <sup>2</sup> /s
<b>TRANSFER_GR</b>	growth resp for transfer growth displayed in this timestep	gC/m <sup>2</sup> /s
<b>TRANSFER_LEAF_GR</b>	leaf growth respiration from storage	gC/m <sup>2</sup> /s
<b>TRANSFER_LIVECROOT_GR</b>	live coarse root growth respiration from storage	gC/m <sup>2</sup> /s
<b>TRANSFER_LIVESTEM_GR</b>	live stem growth respiration from storage	gC/m <sup>2</sup> /s
<b>TREFMNAV</b>	daily minimum of average 2-m temperature	K
<b>TREFMNAV_R</b>	Rural daily minimum of average 2-m temperature	K

Name	Long-name	Units
TREFMNAV_U	Urban daily minimum of average 2-m temperature	K
TREFMXAV	daily maximum of average 2-m temperature	K
TREFMXAV_R	Rural daily maximum of average 2-m temperature	K
TREFMXAV_U	Urban daily maximum of average 2-m temperature	K
TSA	2m air temperature	K
TSAI	total projected stem area index	none
TSA_R	Rural 2m air temperature	K
TSA_U	Urban 2m air temperature	K
TSOI	soil temperature	K
TSOI_10CM	soil temperature in top 10cm of soil	K
TV	vegetation temperature	K
TV24	vegetation temperature (last 24hrs)	K
TV240	vegetation temperature (last 240hrs)	K
Tair	atmospheric air temperature	K
U10	10-m wind	m/s
U10_DUST	10-m wind for dust model	m/s
ULRAD	upward longwave radiation above the canopy	W/m^2
URBAN_AC	urban air conditioning flux	watt/m^2
URBAN_HEAT	urban heating flux	watt/m^2
VA	atmospheric wind speed plus convective velocity	m/s
VCMXSHA	shaded leaf Vcmax	umolCO2/m^2/s
VCMXSUN	sunlit leaf Vcmax	umolCO2/m^2/s
VEGFIRE	pft-level fire loss	gC/m^2/s
VOCFLXT	total VOC flux into atmosphere	uGC/M2/H
VOLR	RTM storage: LIQ	m3
VOLR_ICE	RTM storage: ICE	m3
WA	water in the unconfined aquifer	mm
WASTEHEAT	sensible heat flux from heating/cooling sources of urban waste heat	watt/m^2
WATDRY	watdry for entire column	mm3/mm3

Name	Long-name	Units
<b>WATOPT</b>	watopt for entire column	mm3/mm3
<b>WF</b>	soil water as frac. of whc for top 0.5 m	proportion
<b>WIND</b>	atmospheric wind velocity magnitude	m/s
<b>WLIM</b>	Water limitation used in bgmoist (atmp factor)	
<b>WOODC</b>	wood C	gC/m^2
<b>WOODC</b>	wood C	gC/m2
<b>WOODC_ALLOC</b>	wood C allocation	gC/m^2/s
<b>WOODC_ALLOC</b>	wood C allocation	gC/m2/s
<b>WOODC_LOSS</b>	wood C loss	gC/m^2/s
<b>WOODC_LOSS</b>	wood C loss	gC/m2/s
<b>WOOD_HARVESTC</b>	wood harvest (to product pools)	gC/m^2/s
<b>WOOD_HARVESTN</b>	wood harvest (to product pools)	gN/m^2/s
<b>WT</b>	total water storage (unsaturated soil water + groundwater)	mm
<b>Wind</b>	atmospheric wind velocity magnitude	m/s
<b>XSCPOOL</b>	total excess Carbon	g/m2
<b>XSMRPOOL</b>	temporary photosynthate C pool	gC/m^2
<b>XSMRPOOL_C13RATIO</b>	C13/C(12+13) ratio for xsmrpool	proportion
<b>XSMRPOOL_RECOVER</b>	C flux assigned to recovery of negative xsmrpool	gC/m^2/s
<b>Z0HG</b>	roughness length over ground, sensible heat	m
<b>Z0HV</b>	roughness length over vegetation, sensible heat	m
<b>Z0M</b>	momentum roughness length	m
<b>Z0MG</b>	roughness length over ground, momentum	m
<b>Z0MV</b>	roughness length over vegetation, momentum	m
<b>Z0QG</b>	roughness length over ground, latent heat	m
<b>Z0QV</b>	roughness length over vegetation, latent heat	m
<b>ZBOT</b>	atmospheric reference height	m
<b>ZII</b>	convective boundary height	m
<b>ZWT</b>	water table depth	m

Name	Long-name	Units
aais_area	Antarctic ice area	km <sup>2</sup>
aais_mask	Antarctic mask	unitless
gris_area	Greenland ice area	km <sup>2</sup>
gris_mask	Greenland mask	unitless

## Examples of using different namelist features

Below we will give examples of user namelists that activate different commonly used namelist features. We will discuss the namelist features in different examples and then show a user namelist that includes an example of the use of these features. First we will show the default namelist that doesn't activate any user options.

### The default namelist

Here we give the default namelist as it would be created for a I1850CN compset at 0.9x1.25 resolution with a gx1v6 land-mask. To edit the namelist you would edit the BuildConf/clm.buildnml.csh under your case (or before **configure** include a user namelist with just the items you want to change). For simplicity we will just show the namelist and NOT the entire file. In the sections below, for simplicity we will just show the user namelist (`user_nl_clm`) that will add (or modify existing) namelist items to the namelist. Again, just adding the `user_nl_clm` file to your case directory, before "**configure -case**" is invoked will cause the given namelist items to appear in your CLM namelist.

### Example 1-2. Default CLM Namelist

```

&clm_inparm
  co2_ppmv    = 284.7
  co2_type    = 'constant'
  create_crop_landunit = .false.
  dtime      = 1800
  fatmgrid   = '$DIN_LOC_ROOT/lnd/clm2/griddata/griddata_0.9x1.25_070212.nc'
  fatmlndfrc =
'$DIN_LOC_ROOT/lnd/clm2/griddata/fracdata_0.9x1.25_gx1v6_c090317.nc'
  finidat    = 'I1850CN_f09_g16_c100503.clm2.r.0001-01-01-00000.nc'
  fpftcon    = '$DIN_LOC_ROOT/lnd/clm2/pftdata/pft-physiology.c110425.nc'
  frivinp_rtm = '$DIN_LOC_ROOT/lnd/clm2/rtmdata/rdirc_0.5x0.5_simyr2000_c101124.nc'
  fsnowaging =
'$DIN_LOC_ROOT/lnd/clm2/snicalldata/snical_drdt_bst_fit_60_c070416.nc'
  fsnowoptics =
'$DIN_LOC_ROOT/lnd/clm2/snicalldata/snical_optics_5bnd_c090915.nc'
  fsurdat    =
'$DIN_LOC_ROOT/lnd/clm2/surfdata/surfdata_0.9x1.25_simyr1850_c091006.nc'
  ice_runoff  = .true.
  outnc_large_files = .true.
  rtm_nsteps  = 6
  urban_hac   = 'ON_WASTEHEAT'
  urban_traffic = .false.
/
&ndepdyn_nml
  stream_fldfilename_ndep =
'$DIN_LOC_ROOT/lnd/clm2/ndepdata/frndep_clm_hist_simyr1849-2006_1.9x2.5_c100428.nc'
  stream_year_first_ndep = 1850
  stream_year_last_ndep = 1850
/

```

Note that the namelist introduces some of the history namelist options that will be talked about in further detail below (`hist_mfilt` and `hist_nhtfrq`).

## Adding/removing fields on your primary history file

The primary history files are output monthly, and contain an extensive list of fieldnames, but the list of fieldnames can be added to using `hist_fincl1` or removed from by adding fieldnames to `hist_fexcl1`. A sample user namelist `user_nl_clm` adding few new fields (cosine of solar zenith angle, and solar declination) and excluding a few standard fields is (ground temperature, vegetation temperature, soil temperature and soil water).:

### Example 1-3. Example `user_nl_clm` namelist adding and removing fields on primary history file

```
&clm_inparm
hist_fincl1 = 'COSZEN', 'DECL'
hist_fexcl1 = 'TG', 'TV', 'TSOI', 'H2OSOI'
/
```

## Adding auxiliary history files and changing output frequency

The `hist_fincl2` through `hist_fincl6` set of namelist variables add given history fieldnames to auxiliary history file "streams", and `hist_fexcl2` through `hist_fexcl6` set of namelist variables remove given history fieldnames from history file auxiliary "streams". A history "stream" is a set of history files that are produced at a given frequency. By default there is only one stream of monthly data files. To add more streams you add history fieldnames to `hist_fincl2` through `hist_fincl6`. The output frequency and the way averaging is done can be different for each history file stream. By default the primary history files are monthly and any others are daily. You can have up to six active history streams, but you need to activate them in order. So if you activate stream "6" by setting `hist_fincl6`, but if any of `hist_fincl2` through `hist_fincl5` are unset, only the history streams up to the first blank one will be activated.

The frequency of the history file streams is given by the namelist variable `hist_nhtfrq` which is an array of rank six for each history stream. The values of the array `hist_nhtfrq` must be integers, where the following values have the given meaning:

*Positive value* means the output frequency is the number of model steps between output.

*Negative value* means the output frequency is the absolute value in hours given (i.e -1 would mean an hour and -24 would mean a day).

*Zero* means the output frequency is monthly. This is the default for the primary history files.

The number of samples on each history file stream is given by the namelist variable `hist_mfilt` which is an array of rank six for each history stream. The values of the array `hist_mfilt` must be positive integers. By default the primary history file stream has one time sample on it (i.e. output is to separate monthly files), and all other streams have thirty time samples on them.

A sample user namelist `user_nl_clm` turning on four extra file streams for output: daily, six-hourly, hourly, and every time-step, leaving the primary history files as monthly, and changing the number of samples on the streams to: yearly (12), thirty, weekly (28), daily (24), and daily (48) is:

**Example 1-4. Example `user_nl_clm` namelist adding auxiliary history files and changing output frequency**

```
&clm_inparm
hist_fincl2 = 'TG', 'TV'
hist_fincl3 = 'TG', 'TV'
hist_fincl4 = 'TG', 'TV'
hist_fincl5 = 'TG', 'TV'
hist_nhtfrq = 0, -24, -6, -1, 1
hist_mfilt = 12, 30, 28, 24, 48
/
```

## Removing all history fields

Sometimes for various reasons you want to remove all the history fields either because you want to do testing without any output, or you only want a very small custom list of output fields rather than the default extensive list of fields. By default only the primary history files are active, so technically using `hist_fexcl1` explained in the first example, you could list *ALL* of the history fields that are output in `hist_fexcl1` and then you wouldn't get any output. However, as the list is very extensive this would be a cumbersome thing to do. So to facilitate this `hist_empty_htapes` allows you to turn off all default output. You can still use `hist_fincl1` to turn your own list of fields on, but you then start from a clean slate. A sample user namelist `user_nl_clm` turning off all history fields and then activating just a few selected fields (ground and vegetation temperatures and absorbed solar radiation) is:

**Example 1-5. Example `user_nl_clm` namelist removing all history fields**

```
&clm_inparm
hist_empty_htapes = .true.
hist_fincl1 = 'TG', 'TV', 'FSA'
/
```

Note, you could also build adding the "-noio" option to `CLM_CONFIG_OPTS`. But, this would build the model without history output and you wouldn't be able to add that in later.

## Various ways to change history output averaging flags

There are two ways to change the averaging of output history fields. The first is using `hist_avgflag_pertape` which gives a default value for each history stream, the second is when you add fields using `hist_fincl*`, you add an averaging flag to the end of the field name after a colon (for example 'TSOI:X', would output the maximum of TSOI). The types of averaging that can be done are:

- A Average, over the output interval.
- I Instantaneous, output the value at the output interval.
- X Maximum, over the output interval.
- M Minimum, over the output interval.

The default averaging depends on the specific fields, but for most fields is an average. A sample user namelist `user_nl_clm` making the monthly output fields all averages (except TSOI for the first two streams and FIRE for the 5th stream), and adding auxiliary file streams for instantaneous (6-hourly),

maximum (daily), minimum (daily), and average (daily). For some of the fields we diverge from the per-tape value given and customize to some different type of optimization.

**Example 1-6. Example `user_nl_clm` namelist with various ways to average history fields**

```
&clm_inparm
hist_empty_htapes = .true.
hist_fincl1 = 'TSOI:X', 'TG', 'TV', 'FIRE', 'FSR', 'FSH',
              'EFLX_LH_TOT', 'WT'
hist_fincl2 = 'TSOI:X', 'TG', 'TV', 'FIRE', 'FSR', 'FSH',
              'EFLX_LH_TOT', 'WT'
hist_fincl3 = 'TSOI', 'TG:I', 'TV', 'FIRE', 'FSR', 'FSH',
              'EFLX_LH_TOT', 'WT'
hist_fincl4 = 'TSOI', 'TG', 'TV:I', 'FIRE', 'FSR', 'FSH',
              'EFLX_LH_TOT', 'WT'
hist_fincl5 = 'TSOI', 'TG', 'TV', 'FIRE:I', 'FSR', 'FSH',
              'EFLX_LH_TOT', 'WT'
hist_avgflag_pertape = 'A', 'I', 'X', 'M', 'A'
hist_nhtfrq = 0, -6, -24, -24, -24
/
```

In the example we put the same list of fields on each of the tapes: soil-temperature, ground temperature, vegetation temperature, emitted longwave radiation, reflected solar radiation, sensible heat, total latent-heat, and total water storage. We also modify the soil-temperature for the primary and secondary auxiliary tapes by outputting them for a maximum instead of the prescribed per-tape of average and instantaneous respectively. For the tertiary auxiliary tape we output ground temperature instantaneous instead of as a maximum, and for the fourth auxiliary tape we output vegetation temperature instantaneous instead of as a minimum. Finally, for the fifth auxiliary tapes we output FIRE instantaneously instead of as an average.

**Note:** We also use `hist_empty_htapes` as in the previous example, so we can list ONLY the fields that we want on the primary history tapes.

## **Outputting history files as a vector in order to analyze the plant function types within gridcells**

By default the output to history files are the grid-cell average of all land-units, and vegetation types within that grid-cell, and output is on the full 2D latitude/longitude grid with ocean masked out. Sometimes it's important to understand how different land-units or vegetation types are acting within a grid-cell. The way to do this is to output history files as a 1D-vector of all land-units and vegetation types. In order to display this, you'll need to do extensive post-processing to make sense of the output. Often you may only be interested in a few points, so once you figure out the 1D indices for the grid-cells of interest, you can easily view that data. 1D vector output can also be useful for single point datasets, since it's then obvious that all data is for the same grid cell.

To do this you use `hist_dov2xy` which is an array of rank six for each history stream. Set it to `.false.` if you want one of the history streams to be a 1D vector. You can also use `hist_typeid_pertape` if you want to average over all the: Plant-Function-Types, columns, land-units, or grid-cells. A sample user namelist `user_nl_clm` leaving the primary monthly files as 2D, and then

doing grid-cell (GRID), column (COLS), and no averaging over auxiliary tapes output daily for a single field (ground temperature) is:

**Example 1-7. Example `user_nl_clm` namelist outputting some files in 1D Vector format**

```
&clm_inparm
hist_fincl2 = 'TG'
hist_fincl3 = 'TG'
hist_fincl4 = 'TG'
hist_fincl5 = 'TG'
hist_fincl6 = 'TG'
hist_dov2xy = .true., .false., .false., .false.
hist_type2d_pertape = ' ', 'GRID', 'COLS', ''
hist_nhtfrq = 0, -24, -24, -24
/
```

**Warning**

LAND and COLS are also options to the pertape averaging, but currently there is a bug with them and they fail to work.

**Note:** Technically the default for `hist_nhtfrq` is for primary files output monthly and the other auxiliary tapes for daily, so we don't actually have to include `hist_nhtfrq`, we could use the default for it. Here we specify it for clarity.

**Caution**

Visualizing global 1D vector files will take effort. You'll probably want to do some post-processing and possibly just extract out single points of interest to see what is going on. Since, the output is a 1D vector, of only land-points traditional plots won't be helpful. The number of points per grid-cell will also vary for anything, but grid-cell averaging. You'll need to use the output fields `pfts1d_ixy`, and `pfts1d_jxy`, to get the mapping of the fields to the global 2D array. `pfts1d_itype_veg` gives you the PFT number for each PFT. Most likely you'll want to do this analysis in a data processing tool (such as NCL, Matlab, Mathematica, IDL, etcetera that is able to read and process NetCDF data files).

## Conclusion to namelist examples

We've given various examples of namelists that feature the use of different namelist options to customize a case for particular uses. Most the examples revolve around how to customize the output history fields. This should give you a good basis for setting up your own CLM namelist.

## Customizing the DATM Namelist and Streams files

When running "I" compsets with CLM you use the DATM model to give atmospheric forcing data to CLM. There are four ways to customize DATM:

1. *DATM Main Namelist* (`datm_in`)
2. *DATM Stream Namelist* (`datm_atm_in`)
3. *DATM stream files*
4. *DATM template file* (`Tools/Templates.datm.cpl7.template`)

### The Data Model Documentation

([http://www.cesm.ucar.edu/models/cesm1.0/data8/data8\\_doc/book1.html](http://www.cesm.ucar.edu/models/cesm1.0/data8/data8_doc/book1.html)) gives the details of all the options for the data models and for DATM specifically. It goes into detail on all namelist items both for DATM and for DATM streams. It shows examples of stream files and talks about their use. In Appendix A we talk about editing the CLM and DATM template files. So here we won't talk about the DATM template file, and we won't list ALL of the DATM namelist options, nor go into great details about stream files. But, we will talk about a few of the different options that are relevant for running with CLM. All of the options for changing the namelists or stream files is done by editing the `Buildconf/datm.buildnml.csh` file.

Because, they aren't useful for work with CLM we will NOT discuss any of the options for the main DATM namelist. Use the DATM Users Guide at the link above to find details of that. For the streams namelist we will discuss three items:

1. *mapalgo*
2. *taxmode*
3. *tintalgo*

And for the streams file itself we will discuss:

### *offset*

Again everything else (and including the above items) are discussed in the Data Model User's Guide. Of the above the last three: offset, taxmode and tintalgo are all closely related and have to do with the time interpolation of the DATM data.

### mapalgo

`mapalgo` sets the spatial interpolation method to go from the DATM input data to the output DATM model grid. The default is `bilinear`. For CLM1PT we set it to `nn` to just select the nearest neighbor. This saves time and we also had problems running the interpolation for single-point mode.

### taxmode

`taxmode` is the time axis mode. For CLM we usually have it set to `cycle` which means that once the end of the data is reached it will start over at the beginning. The `extend` modes is used have it use the last time-step of the forcing data once it reaches the end of forcing data (or use the first time-step before it reaches where the forcing data starts). See the warning below about the `extend`

mode.

## Warning

*THE `extend` OPTION NEEDS TO BE USED WITH CAUTION!* It is only invoked by default for the CLM1PT mode and is only intended for the supported urban datasets to extend the data for a single time-step. If you have the model *run extensively through periods in this mode you will effectively be repeating that last time-step over that entire period*. This means the output of your simulation will be worthless.

offset (in the stream file)

`offset` is the time offset in seconds to give to each stream of data. Normally it is NOT used because the time-stamps for data is set correctly for each stream of data. Note, the `offset` may NEED to be adjusted depending on the `taxmode` described above, or it may need to be adjusted to account for data that is time-stamped at the END of an interval rather than the middle or beginning of interval. The `offset` can be set in the stream file rather than on the stream namelist. For data with a `taxmode` method of `coszen` the time-stamp needs to be for the beginning of the interval, while for other data it should be the midpoint. The `offset` can be used to adjust the time-stamps to get the data to line up correctly.

`tintalgo`

`tintalgo` is the time interpolation algorithm. For CLM we usually use one of three modes: `coszen`, `nearest`, or `linear`. We use `coszen` for solar data, `nearest` for precipitation data, and `linear` for everything else. If your data is half-hourly or hourly, `nearest` will work fine for everything. The `coszen` scaling is useful for longer periods (three hours or more) to try to get the solar to match the cosine of the solar zenith angle over that longer period of time. If you use `linear` for longer intervals, the solar will cut out at night-time anyway, and the straight line will be a poor approximation of the cosine of the solar zenith angle of actual solar data. `nearest` likewise would be bad for longer periods where it would be much higher than the actual values.

**Note:** For `coszen` the time-stamps of the data should correspond to the beginning of the interval the data is measured for. Either make sure the time-stamps on the datafiles is set this way, or use the `offset` described above to set it.

**Note:** For `nearest` and `linear` the time-stamps of the data should correspond to the middle of the interval the data is measured for. Either make sure the time-stamps on the datafiles is set this way, or use the `offset` described above to set it.

In the sections below we go over each of the relevant DATM\_MODE options and what the above DATM settings are for each. This gives you examples of actual usage for the settings. We also describe in what ways you might want to customize them for your own case.

## CLM\_QIAN mode and it's DATM settings

In `CLM_QIAN` mode the Qian dataset is used which has 6-hourly solar and precipitation data, and 3-hourly for everything else. The dataset is divided into those three data streams: solar, precipitation, and everything else (temperature, pressure, humidity and wind). The time-stamps of the data were also adjusted so that they are the beginning of the interval for solar, and the middle for the other two. Because, of this the `offset` is set to zero, and the `tintalgo` is: `coszen`, `nearest`, and `linear` for the solar, precipitation and other data respectively. `taxmode` is set to `cycle` and `mapalgo` is set to `bilinear` so that the data is spatially interpolated from the input T62 grid to the grid the atmosphere model is being run at.

Normally you wouldn't customize the `CLM_QIAN` settings, but you might replicate it's use for your own global data that had similar temporal characteristics.

## CLM1PT mode and it's DATM settings

In `CLM1PT` mode the model is assumed to have half-hourly or hourly data for a single-point. For the supported datasets that is exactly what it has. But, if you add your own data you may need to make adjustments accordingly. Using the `CLM_USRDAT_NAME` option you can easily extend this mode for your own datasets that may be regional or even global and could be at different temporal frequencies. If you do so you'll need to make adjustments to your DATM settings. The dataset has all data in a single stream file. The time-stamps of the data were also adjusted so that they are at the middle of the interval. Because, of this the `offset` is set to zero, and the `tintalgo` is set to `nearest`. `taxmode` is set to `extend` and `mapalgo` is set to `nn` so that simply the nearest point is used.

If you are using your own data for this mode and it's not at least hourly you'll want to adjust the DATM settings for it. If the data is three or six hourly, you'll need to divide it up into separate streams like in `CLM_QIAN` mode which will require fairly extensive changes to the DATM namelist and streams files. For an example of doing this see Example 5-7.

## CPLHIST3HrWx mode and it's DATM settings

In `CPLHIST3HrWx` mode the model is assumed to have 3-hourly for a global grid from a previous CESM simulation. Like `CLM_QIAN` mode the data is divided into three streams: one for precipitation, one for solar, and one for everything else. The time-stamps for Coupler history files for CESM is at the end of the interval, so the offset needs to be set in order to adjust the time-stamps to what it needs to be for the `tintalgo` settings. For precipitation `taxmode` is set to `nearest` so the `offset` is set to -5400 seconds so that the ending time-step is adjusted by an hour and half to the middle of the interval. For solar `taxmode` is set to `coszen` so the `offset` is set to -10800 seconds so that the ending time-step is adjust by three hours to the beginning of the interval. For everything else `taxmode` is set to `linear` so the `offset` is set to -5400 seconds so that the ending time-step is adjusted by an hour and half to the middle of the interval.

Normally you wouldn't modify the DATM settings for this mode. However, if you had data at a different frequency than 3-hours you would need to modify the `offset` and possibly the `taxmode`. The other two things that you might modify would be the path to the data (which you can change in the DATM template see Appendix A) or the domain file for the resolution (which is currently hardwired to f09). For data at a

different input resolution you would need to change the domain file in the streams file to use a domain file to the resolution that the data comes in on.

## Conclusion to customizing chapter

We've given extensive details on customizing cases with CLM, by choosing compsets, by changing **configure** options and interacting with the CLM "**configure**" and "**build-namelist**" scripts, we've given details on all of the CLM namelist items, and finally given some instruction in customizing the DATM namelist and streams files. In the next chapter we talk about further ways to customize cases with CLM by creating your own datasets using the tools provided in CLM.

# Chapter 2. Using the CLM tools to create your own input datasets

There are several tools provided with CLM that allow you to create your own input datasets at resolutions you choose, or to interpolate initial conditions to a different resolution, or used to compare CLM history files between different cases. The tools are all available in the `models/lnd/clm/tools` directory. Most of the tools are FORTRAN stand-alone programs in their own directory, but there is also a suite of NCL scripts in the `ncl_scripts` directory. Some of the NCL scripts are very specialized and not meant for general use, and we won't document them here. They still contain documentation in the script itself and the README file in the tools directory. But, the list of generally important scripts and programs are:

1. `cprnc` to compare NetCDF files with a time axis.
2. `interpinic` to interpolate initial condition files.
3. `mkgriddata` to create grid datasets.
4. `mkgdatabuild` to create domain files from grid datasets used by DATM or docn.
5. `mksurfdatal` to create surface datasets from grid datasets.
6. `ncl_scripts/getregional_datasets.pl` script to extract a region or a single-point from global input datasets. See the single-point chapter for more information on this.
7. `ncl_scripts/npdepregrid.ncl` interpolate the Nitrogen deposition datasets to a new resolution.
8. `ncl_scripts/aerdepregrid.ncl` interpolate the Aerosol deposition datasets to a new resolution.

In the sections to come we will go into detailed description of how to use each of these tools in turn. First, however we will discuss the common environment variables and options that are used by all of the FORTRAN tools. Second, we go over the outline of the entire file creation process for all input files needed by CLM for a new resolution, then we turn to each tool. In the last section we will discuss how to customize files for particular observational sites.

## Common environment variables and options used in building the FORTRAN tools

The FORTRAN tools all have similar makefiles, and similar options for building. All of the Makefiles use GNU Make extensions and thus require that you use GNU make to use them. They also auto detect the type of platform you are on, using "uname -s" and set the compiler, compiler flags and such accordingly. There are also environment variables that can be set to set things that must be customized. All the tools use NetCDF and hence require the path to the NetCDF libraries and include files. On some platforms (such as Linux) multiple compilers can be used, and hence there are env variables that can be set to change the FORTRAN and/or "C" compilers used. The tools other than `cprnc` also allow finer control, by also allowing the user to add compiler flags they choose, for both FORTRAN and "C", as well as picking the compiler, linker and and add linker options. Finally the tools other than `cprnc` allow

you to turn optimization on (which is off by default but on for the **mksurfdata** and **interpinic** programs) with the OPT flag so that the tool will run faster. To get even faster performance, the **interpinic**, **mksurfdata**, and **mkgriddata** programs allow you to also use the SMP to turn on multiple shared memory processors. When SMP=TRUE you set the number of threads used by the program with the OMP\_NUM\_THREADS environment variable.

Options used by all: **cprnc**, **interpinic**, **mkdatadomain**, **mkgriddata**, and **mksurfdata**

**LIB\_NETCDF** -- sets the location of the NetCDF library.

**INC\_NETCDF** -- sets the location of the NetCDF include files.

**USER\_FC** -- sets the name of the FORTRAN compiler.

Options used by: **interpinic**, **mkdatadomain**, **mkgriddata**, and **mksurfdata**

**MOD\_NETCDF** -- sets the location of the NetCDF FORTRAN module.

**USER\_LINKER** -- sets the name of the linker to use.

**USER\_CPPDEFS** -- adds any CPP defines to use.

**USER\_CFLAGS** -- add any "C" compiler flags to use.

**USER\_FFLAGS** -- add any FORTRAN compiler flags to use.

**USER\_LDFLAGS** -- add any linker flags to use.

**USER\_CC** -- sets the name of the "C" compiler to use.

**OPT** -- set to TRUE to compile the code optimized (TRUE or FALSE)

Options used by: **interpinic**, **mkgriddata**, and **mksurfdata**:

**SMP** -- set to TRUE to turn on shared memory parallelism (i.e. OpenMP) (TRUE or FALSE)

**Filepath** -- list of directories to build source code from.

**Srcfiles** -- list of source code filenames to build executable from.

Options used only by **cprnc**:

**EXEDIR** -- sets the location where the executable will be built.

**VPATH** -- colon delimited path list to find the source files.

More details on each environment variable.

#### **LIB\_NETCDF**

This variable sets the path to the NetCDF library file (`libnetcdf.a`). If not set it defaults to `/usr/local/lib`. In order to use the tools you need to build the NetCDF library and be able to link to it. In order to build the model with a particular compiler you may have to compile the NetCDF library with the same compiler (or at least a compatible one).

#### **INC\_NETCDF**

This variable sets the path to the NetCDF include directory (in order to find the include file `netcdf.inc`). if not set it defaults to `/usr/local/include`.

#### **MOD\_NETCDF**

This variable sets the path to the NetCDF module directory (in order to find the NetCDF FORTRAN-90 module file when NetCDF is used with a FORTRAN-90 **use statement**). When not set it defaults to the **LIB\_NETCDF** value.

#### **USER\_FC**

This variable sets the command name to the FORTRAN-90 compiler to use when compiling the tool. The default compiler to use depends on the platform. And for example, on the AIX platform this variable is NOT used

#### USER\_LINKER

This variable sets the command name to the linker to use when linking the object files from the compiler together to build the executable. By default this is set to the value of the FORTRAN-90 compiler used to compile the source code.

#### USER\_CPPDEFS

This variable adds additional optional values to define for the C preprocessor. Normally, there is no reason to do this as there are very few CPP tokens in the CLM tools. However, if you modify the tools there may be a reason to define new CPP tokens.

#### USER\_CC

This variable sets the command name to the "C" compiler to use when compiling the tool. The default compiler to use depends on the platform. And for example, on the AIX platform this variable is NOT used

#### USER\_CFLAGS

This variable adds additional compiler options for the "C" compiler to use when compiling the tool. By default the compiler options are picked according to the platform and compiler that will be used.

#### USER\_FFLAGS

This variable adds additional compiler options for the FORTRAN-90 compiler to use when compiling the tool. By default the compiler options are picked according to the platform and compiler that will be used.

#### USER\_LDFLAGS

This variable adds additional options to the linker that will be used when linking the object files into the executable. By default the linker options are picked according to the platform and compiler that is used.

#### SMP

This variable flags if shared memory parallelism (using iOpenMP) should be used when compiling the tool. It can be set to either TRUE or FALSE, by default it is set to FALSE, so shared memory parallelism is NOT used. When set to TRUE you can set the number of threads by using the OMP\_NUM\_THREADS environment variable. Normally, the most you would set this to would be to the number of on-node CPU processors. Turning this on should make the tool run much faster.

#### **Caution**

Note, that depending on the compiler answers may be different when SMP is activated.

#### OPT

This variable flags if compiler optimization should be used when compiling the tool. It can be set to either TRUE or FALSE, by default it is set to FALSE for **mkdatadomain** and TRUE for **mksurfdata** and **interpinic**. Turning this on should make the tool run much faster.

## Caution

Note, you should expect that answers will be different when OPT is activated.

### Filepath

All of the tools are stand-alone and don't need any outside code to operate. The `Filepath` is the list of directories needed to compile and hence is always simply `."` the current directory. Several tools use copies of code outside their directory that is in the CESM distribution (either `csm_share` code or CLM source code).

### Srcfiles

The `Srcfiles` lists the filenames of the source code to use when building the tool.

### EXEDIR

The `cprnc` tool uses this variable to set the location of where the executable will be built. The default is the current directory.

### VPATH

The `cprnc` tool uses this variable to set the colon delimited pathnames of where the source code exists. The default is the current directory.

**Note:** There are several files that are copies of the original files from either `models/lnd/clm/src/main`, `models/csm_share/shr`, or copies from other tool directories. By having copies the tools can all be made stand-alone, but any changes to the originals will have to be put into the tool directories as well.

The `README.filecopies` (which can be found in `models/lnd/clm/tools`) is repeated here.

`models/lnd/clm/tools/README.filecopies` May/26/2011

There are several files that are copies of the original files from either `models/lnd/clm/src/main`, `models/csm_share/shr`, or copies from other tool directories. By having copies the tools can all be made stand-alone, but any changes to the originals will have to be put into the tool directories as well.

#### I. Files that are IDENTICAL:

1. `csm_share` files copied that should be identical to `models/csm_share/shr`:

```
shr_kind_mod.F90  
shr_const_mod.F90  
shr_log_mod.F90  
shr_timer_mod.F90  
shr_string_mod.F90  
shr_file_mod.F90
```

2. `clm/src` files copied that should be identical to `models/lnd/clm/src/main`:

```
clm_varctl.F90  
nanMod.F90
```

```
3. Files shared between mkgridata and mksurfdata that are identical:  
(these all came from a much older version of clm)  
  
    ncdioMod.F90  
    areaMod.F90  
    mkvarpar.F90  
  
II. Files with differences  
  
1. csm_share files copied with differences:  
  
    shr_sys_mod.F90 - Remove mpi abort and reference to shr_mpi_mod.F90.  
  
2. clm/src files with differences:  
  
    fileutils.F90 --- Remove use of masterproc and spmdMod and endrun in abortutils.  
  
3. Files shared between mkgridata and mksurfdata different from models/lnd/clm/src:  
  
    domainMod.F90 ---- Highly customized based off an earlier version of clm code.  
                      Remove use of abortutils, spmdMod. clm version uses latlon  
                      this version uses domain in names. Distributed memory  
                      parallelism is removed.
```

## General information on running the FORTRAN tools

The tools run either one of two ways, with a namelist to provide options, or with command line arguments (and NOT both). **interpnic** and **cprnc** run with command line arguments, and the other tools run with namelists.

### Running FORTRAN tools with namelists

**mkgridata**, **mksurfdata** and **mkdatadomain** run with namelists that are read from standard input. Hence, you create a namelist and then run them by redirecting the namelist file into standard input as follows:

```
./program < namelist
```

For programs with namelists there is at least one sample namelist with the name "program".namelist (i.e. mksurfdata.namelist for the **mksurfdata** program). There may also be other sample namelists that end in a different name besides "namelist". Namelists that you create should be similar to the example namelist. The namelist values are also documented along with the other namelists in the:

```
models/lnd/clm/bld/namelist_files/namelist_definition.xml  
(..../bld/namelist_files/namelist_definition.xml) file and default values in the:  
models/lnd/clm/bld/namelist_files/namelist_defaults_clm_tools.xml  
(..../bld/namelist_files/namelist_defaults_clm_tools.xml) file.
```

### Running FORTRAN tools with command line options

**interpnic** and **cprnc** run with command line arguments. The detailed sections below will give you more information on the command line arguments specific to each tool. Also running the tool without any

arguments will give you a general synopsis on how to run the tool. For example to get help on running **interpinic** do the following.

```
cd models/lnd/clm/tools/interpinic  
gmake  
../interpinic
```

## Running FORTRAN tools built with SMP=TRUE

When you enable SMP=TRUE on your build of one of the tools that make use of it, you are using OpenMP for shared memory parallelism (SMP). In SMP loops are run in parallel with different threads run on different processors all of which access the same memory (called on-node). Thus you can only usefully run up to the number of processors that are available on a single-node of the machine you are running on. For example, on the NCAR machine bluefire there are 32 processors per node, but the SMT hardware on the machine allows you to submit twice as many threads or 64 threads. So to run the **mksurfdata** on bluefire optimized, with 64 threads you would do the following:

```
cd models/lnd/clm/tools/mksurfdata  
gmake OPT=TRUE SMP=TRUE  
setenv OMP_NUM_THREADS 64  
../mksurfdata < mksurfdata.namelist
```

## The File Creation Process

When just creating a replacement file for an existing one, the relevant tool should be used directly to create the file. When you are creating a set of files for a new resolution there are some dependencies between the tools that you need to keep in mind when creating them. The main dependency is that the **mkgriddata** MUST be done first as the grid dataset is then input into the other tools. Also look at Table 3-1.

### Creating a complete set of files for input to CLM

1. Create grid and fraction datasets

First use **mkgriddata** to create grid and fraction datasets. See the Section called *Using mkgriddata to create grid datasets* for more information on this.

2. Create domain dataset (if NOT already done)

Next use **mkdatadomain** to create a domain file for use by DATM from the grid and fraction datasets just created. This is required, unless a domain file already created was input into **mkgriddata** on the previous step. See the Section called *Using mkdatadomain to create domain datasets for DATM or docn from CLM grid datasets* for more information on this.

3. Create surface datasets

Next use **mksurfdta** to create a surface dataset, using the grid dataset as input. See the Section called *Using mksurfdta to create surface datasets from grid datasets* for more information on this.

4. Interpolate aerosol deposition datasets (optional)

By default the atmosphere model will interpolate these datasets on the fly, so you don't normally need to do this step. A reason you might want to do this is to make the read and interpolation faster, by reducing the amount of data read in and removing the need for the interpolation. So, if you do, you can use **aerdepregrid.ncl** to regrid aerosol deposition datasets to your new resolution using the grid dataset as input. See the Section called *Using aerdepregrid.ncl to interpolate Aerosol deposition datasets* for more information on this.

5. Interpolate Nitrogen deposition datasets (optional, but only needed if running CLMCN)

By default Nitrogen deposition is read in from stream files at 2-degree resolution and interpolated to the resolution you are running at, so you don't need to do this step. As with aerosol deposition datasets a reason you might want to do this is to make the read and interpolation faster, by reducing the amount of data read in and removing the need for the interpolation. So, if you do you can use **ndepregrid.ncl** to regrid Nitrogen deposition datasets to your new resolution using the grid dataset as input. See the Section called *Using ndepreregid.ncl to interpolate Nitrogen deposition datasets* for more information on this.

6. Create some sort of initial condition dataset

You then need to do one of the following three options to have an initial dataset to start from.

a. Use spinup-procedures to create initial condition datasets

The first option is to do the spinup procedures from arbitrary initial conditions to get good initial datasets. This is the most robust method to use. See the Section called *Spinning up the Satellite Phenology Model (CLMSP spinup)* in Chapter 4, the Section called *Spinning up the biogeochemistry Carbon-Nitrogen Model (CN spinup)* in Chapter 4, or the Section called *Spinning up the Carbon-Nitrogen Dynamic Global Vegetation Model (CNDV spinup)* in Chapter 4 for more information on this.

b. Use **interpinic** to interpolate existing initial condition datasets

The next option is to interpolate from spunup datasets at a different resolution, using **interpinic**. See the Section called *Using interpinic to interpolate initial conditions to different resolutions* for more information on this.

c. Start up from arbitrary initial conditions

The last alternative is to run from arbitrary initial conditions without using any spun-up datasets. This is inappropriate when using CLMCN (bgc=cn or cndv) as it takes a long time to spinup Carbon pools.

**Warning**

This is NOT recommended as many fields in CLM take a long time to equilibrate.

7. Enter the new datasets into the **build-namelist** XML database

The last optional thing to do is to enter the new datasets into the **build-namelist** XML database. See Chapter 3 for more information on doing this. This is optional because the user may enter these files into their namelists manually. The advantage of entering them into the database is so that they automatically come up when you create new cases.

## Using the cprnc tool to compare two history files

**cprnc** is a tool shared by both CAM and CLM to compare two NetCDF history files. It differences every field that has a time-axis that is also shared on both files, and reports a summary of the difference. The summary includes the three largest differences, as well as the root mean square (RMS) difference. It also gives some summary information on the field as well. You have to enter at least one file, and up to two files. With one file it gives you summary information on the file, and with two it gives you information on the differences between the two. At the end it will give you a summary of the fields compared and how many fields were different and how many were identical.

Options:

```
-m = do NOT align time-stamps before comparing  
-v = verbose output  
-ipr  
-jpr  
-kpr
```

See the **cprnc** README (*../../tools/cprnc/README*) file for more details which is repeated here:

```
cprnc README  
-----  
  
cprnc is a generic tool for analyzing a netcdf file or comparing  
two netcdf files.  
  
Quick Start Guide:  
-----  
  
> setenv INC_NETCDF $local_netcdf_include_path  
> setenv LIB_NETCDF $local_netcdf_lib_path  
> gmake  
> ./cprnc file1           # for analyze mode  
> ./cprnc file1 file2    # for compare mode  
  
Usage: cprnc [-v] [-d dimname:start[:count]] file1 [file2]  
-m: Compare each time sample. Default is false, i.e. match "time"  
     coordinate values before comparing  
-v: Verbose output  
-d dimname:start[:count]  
     Print variable values for the specified dimname index subrange.  
  
Users Guide:  
-----  
  
cprnc is a Fortran-90 application. It relies on netcdf version 3 or  
later and uses the f90 netcdf interfaces. It requires a netcdf include  
file and a netcdf library. The Makefile assumes those are in  
/usr/local/include and /usr/local/lib respectively. Users can override  
these defaults by setting environment variables INC_NETCDF and LIB_NETCDF  
to the local values of the netcdf paths. The Makefile has entries for  
IRIX64, AIX, Darwin, Cray X1E, OSF1, SUN, and Linux. On Linux, entries  
exist for pgf90 and lf95 compilers. The Makefile is quite extensible for  
new architectures. gmake should be used to build cprnc. The Makefile  
relies on a hard-wired Depends file.
```

cprnc generates an ascii output file via standard out. It initially summarizes some characteristics of the input file[s]. A compare file is generally 132 characters wide and an analyze file is less than 80 characters wide.

In analyze mode, the output for a field looks like

```
( lon, lat, time, ----)
259200 ( 587, 134, 1) ( 269, 59, 1)
FX1      96369 8.273160400390625E+02 0.000000000000000E+00
avg abs field values: 9.052845920820910E+01
```

and a guide to this information is printed at the top of the file

```
( dim1, dim2, dim3, dim4)
ARRSIZ1 ( indx1, indx2, indx3) file 1
FIELD      NVALID      MAX          MIN
```

The first 10 characters of the field name are identified in the first dozen columns of the third line.  
The first line summarizes the names of the dimensions of the field  
The second line summarizes the indices of the maximum and minimum value of the field for the first three dimensions. If the fourth dimension exists, it's always assumed to be time. Time is handled separately.  
The third line summarizes the number of valid values in the array and the maximum and minimum value over those valid values. Invalid values are values that are identified to be "fill" value.  
The last line summarizes some overall statistics including the average absolute value of the valid values of the field.

In comparison mode, the output (132 chars wide) for a field looks like

```
96369 ( lon, lat, time)
259200 ( 422, 198, 1) ( 203, 186, 1) ( 47, 169, \
1) ( 224, 171, 1)
FIRA      96369 1.466549530029297E+02 -3.922052764892578E+01 1.4E+02 \
-3.037954139709473E+01 1.0E+00 -3.979958057403564E+00
96369 1.321966247558594E+02 -1.603044700622559E+01 \
1.084177169799805E+02 3.982142448425293E+00
259200 ( 156, 31, 1) ( 573, 178, 1) ( \
avg abs field values: 6.778244097051392E+01 rms diff: 1.4E+01 avg rel diff(npos): \
4.6E-02 5.960437961084186E+01 \
digits(ndif): 1.2 worst: 0.0
avg decimal \
```

and a guide to this information is printed at the top of the file

```
NDIFFS ( dim1, dim2, dim3, dim4, ... )
ARRSIZ1 ( indx1, indx2, indx3, ... ) file 1
FIELD      NVALID1      MAX1          MIN1          DIFFMAX  VALUES \
RDIFMAX  VALUES
NVALID2      MAX2          MIN2
ARRSIZ2 ( indx1, indx2, indx3, ... ) file 2
```

The information content is identical to the information in analyze mode with the following additions. Two additional lines are added in the main body. Lines 4 and 5 are identical to line 3 and 2 respectively but are associated with file 2 instead of file 1. In addition, the right hand side of lines 2, 3, and 4 contain information about the maximum difference, the location and values of the maximum difference, the relative difference and the location and values of the maximum relative difference. The last two line summarize some overall statistics including average absolute values of the field on the two files, rms difference, average relative difference, average number of digits that match, and the worst case for the number of digits that match.

At the end of the output file, a summary is presented that looks like

```
SUMMARY of cprnc:
A total number of 119 fields were compared
of which 83 has non-zero differences
including 0 that had inconsistent size/shape
A total number of 0 fields in file 1 were analyzed (non-compare mode)
A total number of 10 fields in file 1 could not be analyzed at all
A total number of 0 fields in file 2 were analyzed (non-compare mode)
```

This summarizes the number of fields that were compared, that differed and the number of fields that could not be compared but were analyzed.

Developers Guide:  
-----

The tool works as follows.

Fields can be analyzed if they are int, float or double and have between 0 and n dimensions

In general, fields that appear on both files are compared. If they are sizes, no difference statistics are computed and only a summary of the fields on the files are presented. If fields only appear on one file, those fields are analyzed.

The unlimited dimension is treated uniquely. In general, for files that have a dimension named "time", the time axes are compared and matching time values on the two files are compared one timestep at a time. Time values that don't match are skipped. To override the matching behaviour, use cprnc -m. In this mode, timestamps are compared in indexical space. In analyze mode, the fields are analyzed one timestamp at a time. In general, if there is a "time" axis, it will be the outer-most loop in the output analysis. In compare mode, fields with a time axis and a timestamp that are not common between the two files are ignored.

**Note:** To compare files with OUT a time axis you can use the **cprnc.ncl** NCL script in models/lnd/clm/tools/ncl\_scripts. It won't give you the details on the differences but will report if the files are identical or different.

## Using interpinic to interpolate initial conditions to different resolutions

"interpinic" is used to interpolate initial conditions from one resolution to another. In order to do the interpolation you must first run CLM to create a restart file to use as the "template" to interpolate into. Running from arbitrary initial conditions (i.e. finidat = '') for a single time-step is sufficient to do this. Make sure the model produces a restart file. You also need to make sure that you setup the same configuration that you want to run the model with, when you create the template file.

Command line options to **interpinic**:

-i = Input filename to interpolate from  
-o = Output interpolated file, and starting template file

There is a sample template file in the models/lnd/clm/tools/interpinic directory and can be used to run interpolate to. However, this file was created with an older version of CLM and hence we actually recommend that you would do a short run with CLM to create a template file to use.

**Example 2-1. Example of running CLM to create a template file for interpinic to interpolate to**

```
> cd scripts
> ./create_newcase -case cr_f10_TmpltI1850CN -res f10_f10 -compset I1850CN \
-mach bluefire
> cd cr_f10_TmpltI1850CN
# Set starting date to end of year
> ./xmlchange -file env_conf.xml -id RUN_STARTDATE -val 1948-12-31
# Set year align to starting year
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_ALIGN -val 1948
# Set to run a cold start
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
# Set to run only a single day, so a restart file will be created on Jan/1/1949
> ./xmlchange -file env_run.xml -id STOP_N -val 1
# Then configure, build and run as normal
> ./configure -case
> ./cr_f10_TmpltI1850CN.bluefire.build
> ./cr_f10_TmpltI1850CN.bluefire.submit
# And copy the resulting restart file to your interpinic directory
> cd ..../models/lnd/clm/tools/interpinic
> cp /ptmp/$LOGIN/cr_f10_TmpltI1850CN/run/cr_f10_TmpltI1850CN.clm2.r.1949-01-01-00000.nc .
```

In the next example we build **interpinic** optimized with shared memory on for 64 threads so that it runs as fast as possible, to interpolate one of the standard 1-degree datasets to the above 10x15 template file that we created.

**Example 2-2. Example of building and running interpinic to interpolate a 1-degree `finidat` dataset to 10x15**

```
> cd models/lnd/clm/tools/interpinic
> gmake OPT=TRUE SMP=TRUE
> env OMP_NUM_THREADS=64 ./interpinic -o cr_f10_TmpltI1850CN.clm2.r.1949-01-01-00000.nc /
-i /fs/cgd/csm/inputdata/ccsm4_init/b40.1850.track1.1deg.006/0863-01-01/b40.1850.track1.1deg.006.clm2.r.0863-01-01-00000.nc
```

**Tip:** Running **interpinic** at high resolution can take a long time, so we recommend that you always build it optimized and with shared memory processing on, to cut down the run time as much as possible.

**Warning**

**interpinic** does NOT work for CNDV (bgc=cndv).

In Appendix B we give a simpler way to run **interpinic** for several standard resolutions at once, with a script to loop over several resolutions. This is useful for CLM developers who need to create many `finidat` files at once.

## Using **mkgriddata** to create grid datasets

**mkgriddata** is used to create grid, fraction, and topography datasets to run CLM at a new resolution. It is typically the first step in creating datasets needed to run CLM at a new resolution (followed by **mksurfdatal**, and then the interpolation programs, **aerdepregrid.ncl**, and **ndepregrid.ncl** when running with CN).

### **mkgriddata** namelist

**mkgriddata** is controlled by a namelist. There are ten different namelist items, and you need to use enough of them so that files will be output. The different types of input datasets contain different input data types, that correspond to the three different types of output files: grid, fraction, and topography. Output files for each of these will only be output if there is input data that correspond to these. If you only have input data for grid locations -- you will only get an output grid file. If you have both grid and fraction data you will get grid and fraction data files. If you also have topography data you will also get topo files.

Namelist options to **mkgriddata** include:

```
mksrf_fnavyoro -- Navy orography file to use for land fraction and surface heights.  
mksrf_frawtopo -- Raw topography file with just surface heights.  
mksrf_fciamfile -- CAM initial conditions file with land-fractions and topography  
mksrf_fclmgrid -- CLM grid file  
mksrf_fccsmdom -- CESM domain file  
mksrf_fciamtopo -- CAM topography file  
mksrf_lsmlon -- number of longitude for regional grid  
mksrf_lsmlat -- number of latitudes for regional grid  
mksrf_edgen -- Northern edge for regional grid  
mksrf_edgee -- Southern edge for regional grid  
mksrf_edges -- Eastern edge for regional grid  
mksrf_edgew -- Western edge for regional grid
```

You need to enter one of the following four options:

mksrf_fnavyoro	- high resolution topo dataset (topo data)
mksrf_lsmlon	- number of longitudes
mksrf_lsmlat	- number of latitudes
mksrf_edgen	- northern edge of grid (degrees)
mksrf_edgee	- eastern edge of grid (degrees)
mksrf_edges	- southern edge of grid (degrees)
mksrf_edgew	- western edge of grid (degrees)

or

```
mksrf_fciamfile - CAM topo file (grid and possibly fraction data)
```

or

```
mksrf_fccsmdom - CESM domain file (both grid, and fraction data)
```

or

```
mksrf_fclmgrid      - CLM grid or surface dataset file (grid data)
```

Note, you can provide more than one of the needed datasets, and the output data will be determined by the datasets according to an order of precedence. The order of precedence for data is as follows:

1. mksrf\_fcamfile
2. mksrf\_fclmgrid
3. mksrf\_fnavyoro
4. mksrf\_fccsmdom

Grid data then will be established by the file with the highest precedence. CESM domain files sometimes have latitudes and longitudes that are "off" from the standard by a small amount. By establishing an order of precedence you can ensure that grid locations exactly match a given standard file, even if the values in the domain file are off from that.

There are three different major modes for using "mkgriddata" to create grid files for CLM:

```
mksrf_fnavyoro -- Navy orography file to use for land fraction and surface heights.  
mksrf_frawtopo -- Raw topography file with just surface heights.  
mksrf_fcamfile -- CAM initial conditions file with land-fractions and topography  
mksrf_fclmgrid -- CLM grid file  
mksrf_fccsmdom -- CESM domain file  
mksrf_fcamtopo -- CAM topography file  
mksrf_lsmlon -- number of longitude for regional grid  
mksrf_lsmlat -- number of latitudes for regional grid  
mksrf_edgen -- Northern edge for regional grid  
mksrf_edgee -- Southern edge for regional grid  
mksrf_edges -- Eastern edge for regional grid  
mksrf_edgew -- Western edge for regional grid
```

You need to enter one of the following four options:

```
mksrf_fnavyoro      - high resolution topo dataset (topo data)  
mksrf_lsmlon        - number of longitudes  
mksrf_lsmlat        - number of latitudes  
mksrf_edgen         - northern edge of grid (degrees)  
mksrf_edgee         - eastern edge of grid (degrees)  
mksrf_edges          - southern edge of grid (degrees)  
mksrf_edgew          - western edge of grid (degrees)
```

or

```
mksrf_fcamfile     - CAM topo file (grid and possibly fraction data)
```

or

```
mksrf_fccsmdom    - CESM domain file (both grid, and fraction data)
```

or

```
mksrf_fclmgrid      - CLM grid or surface dataset file (grid data)
```

Note, you can provide more than one of the needed datasets, and the output data will be determined by the datasets according to an order of precedence. The order of precedence for data is as follows:

1. mksrf\_fciamfile
2. mksrf\_fclmgrid
3. mksrf\_fnavyoro
4. mksrf\_fccsmdom

Grid data then will be established by the file with the highest precedence. CESM domain files sometimes have latitudes and longitudes that are "off" from the standard by a small amount. By establishing an order of precedence you can ensure that grid locations exactly match a given standard file, even if the values in the domain file are off from that.

There are three different major modes for using **mkgriddata** to create grid files for CLM:

- Convert CESM domain files to CLM grid files
- Create single point or regional area grid files
- Convert CAM files to CLM grid files

## Convert CESM domain files to CLM grid files

CESM domain files such as used for DATM, include all the information needed to create CLM grid and fraction files.

### Example 2-3. Example mkgriddata namelist to convert CESM 4x5 domain files to CLM grid files

```
&clmexp
  mksrf_fccsmdom=
  '/fs/cgd/csm/inputdata/lnd/dlnd7/domain.lnd.4x5_gx3v5.060404.nc'
  mksrf_fclmgrid=
  '/fs/cgd/csm/inputdata/lnd/clm2/griddata/griddata_4x5_060404.nc'
  /
```

**Tip:** Notice that in the above example, a CLM grid file is included as well, even though it's not required. The reason for this is to ensure that the latitude and longitudes on the output files exactly match a standard grid file.

## Create single point or regional area grid files

The process to create single-point or regional area CLM grid files is the same. You enter the number of latitudes and longitudes you want on your output file and the extent of the grid: North, East, South and West. You also tell **mkgriddata** that you are entering a "regional" grid and you also enter the standard Navy orography dataset (or your own orography file if desired). For a single point you simply enter "1" for the number of latitudes and longitudes, but you still enter the grid extent (of the single grid cell). Here is a sample regional namelist to create a 5x5 regional grid over the Amazon:

### Example 2-4. Example mkgriddata namelist to create regional grid over Amazon

```
&clmexp
mksrf_fnavyoro=
"/fs/cgd/csm/inputdata/lnd/clm2/rawdata/mksrf_navyoro_20min.c010129.nc"
mksrf_lsmlon = 5
mksrf_lsmlat = 5
mksrf_edgee = 303.75
mksrf_edgew = 286.25
mksrf_edges = -15.
mksrf_edgen = -4.
/
```

### Warning

Currently you can *NOT* have regional grids that straddle both sides of the Greenwich (longitude = zero) line.

**Important:** You should enter longitudes with values from 0 to 360 East.

## Convert CAM files to CLM grid files (deprecated)

Older CAM initial files included all the information needed to create CLM grid files. Newer CAM files no longer include land fraction data. Hence you can use CAM files to give you the grid coordinates, but you need other data to give you the land-mask and topography. Since, CAM files no longer contain the needed information, this option is now deprecated. In most cases you should use one of the other two options.

## Using mkdatadomain to create domain datasets for DATM or docn from CLM grid datasets

"mkdatadomain" is used to convert CLM grid and fraction datasets into domain datasets that can be used by either the "datm" or "docn" models. Most often CLM users will want to convert the grid datasets they just created using **mkgriddata** into domain datasets to be used by DATM for an "I" case.

**mkdatadomain** is controlled by a namelist, and has a very straight forward operation with only four namelist items all of which are required. You specify which output mode you want "datm" or "docn", and then set the input CLM grid and frac datasets, and the output domain file.

**Example 2-5. Example mkdatadomain namelist to create a domain file from CLM frac and grid data files**

```
&domain_nl
  dtype = "datm"
  f_fracdata =
  '/fs/cgd/csm/inputdata/lnd/clm2/griddata/fracdata_4x5_USGS_070110.nc'
  f_griddata =
  '/fs/cgd/csm/inputdata/lnd/clm2/griddata/griddata_4x5_060404.nc'
  f_domain =
  'domain.lnd.fv4x5_USGS.090117.nc'
  /
```

## Using mksurfdatal to create surface datasets from grid datasets

**mksurfdatal** is used to create surface-datasets from grid datasets and raw datafiles at half-degree resolution to produce files that describe the surface characteristics needed by CLM (fraction of grid cell covered by different land-unit types, and fraction for different vegetation types, as well as things like soil color, and soil texture, etc.). To run **mksurfdatal** you can either use the **mksurfdatal.pl** script which will create namelists for you using the **build-namelist** XML database, or you can run it by hand using a namelist that you provide (possibly modeled after an example provided in the `models/lnd/clm/tools/mksurfdatal` directory). The namelist for **mksurfdatal** is sufficiently complex that we recommend using the **mksurfdatal.pl** tool to build them. In the next section we describe how to use the **mksurfdatal.pl** script and the following section gives more details on running **mksurfdatal** by hand and the various namelist input variables to it.

### Running mksurfdatal.pl

The script **mksurfdatal.pl** can be used to run the **mksurfdatal** program for several configurations, resolutions, simulation-years and simulation year ranges. It will create the needed namelists for you and move the files over to your inputdata directory location (and create a list of the files created, and for developers this file is also a script to import the files into the svn inputdata repository). It will also use the **build-namelist** XML database to determine the correct input files to use, and for transient cases it will create the appropriate `mksrf_fdynuse` file with the list of files for each year needed for this case. And in the case of urban single-point datasets (where surface datasets are actually input into **mksurfdatal**) it will do the additional processing required so that the output dataset can be used once again by **mksurfdatal**. Because, it figures out namelist and input files for you, it is recommended that you use this script for creation of standard surface datasets. If you need to create surface datasets for customized cases, you might need to run **mksurfdatal** on its own. But you could use **mksurfdatal.pl** with the "-debug" option to give you a namelist to start from. For help on **mksurfdatal.pl** you can use the "-help" option as below:

```
> cd models/lnd/clm/tools/mksurfdatal
```

```
> mksurldata.pl -help
```

The output of the above command is:

```
SYNOPSIS
  mksurldata.pl [options]
OPTIONS
  -crop                         Add in crop datasets
  -dinlc [or -l]                  Enter the directory location for inputdata
                                  (default /fs/cgd/csm/inputdata)
  -debug [or -d]                  Don't actually run -- just print out what
                                  would happen if ran.
  -dynpft "filename"             Dynamic PFT/harvesting file to use
                                  (rather than create it on the fly)
                                  (must be consistent with first year)
  -exedir "directory"            Directory where mksurldata program is
                                  (by default assume it's in the current directory)
  -glc_nec "number"               Number of glacier elevation classes to use (by default 0)
  -irrig                          If you want to include irrigated crop in the output file.
  -years [or -y]                  Simulation year(s) to run over (by default 1850,2000)
                                  (can also be a simulation year range: i.e. 1850-2000)
  -help [or -h]                   Display this help.
  -nomv                           Don't move the files to inputdata after completion.
  -res   [or -r] "resolution"    Resolution(s) to use for files (by default all).
  -rcp   [or -c] "rep-con-path"  Representative concentration pathway(s) to use for
                                  future scenarios
                                  (by default -999.9, where -999.9 means historical).
  -username "clm_usrdat_name"    CLM user data name to find grid file with.

NOTE: years, res, and rcp can be comma delimited lists.

OPTIONS to override the mapping of the input gridded data with hardcoded input

  -pft_frc "list of fractions"  Comma delimited list of percentages for veg types
  -pft_idx "list of veg index"  Comma delimited veg index for each fraction
  -soil_cly "% of clay"         % of soil that is clay
  -soil_col "soil color"        Soil color (1 [light] to 20 [dark])
  -soil_fmx "soil fmax"        Soil maximum saturated fraction (0-1)
  -soil_snd "% of sand"        % of soil that is sand
```

To run the script with optimized **mksurldata** for a 4x5 degree grid for 1850 conditions, on bluefire you would do the following:

#### **Example 2-6. Example of running mksurldata.pl to create a 4x5 resolution fsurdat for a 1850 simulation year**

```
> cd models/lnd/clm/tools/mksurldata
> gmake
> mksurldata.pl -y 1850 -r 4x5
```

## Running **mksurfdatal** by Hand

In the above section we show how to run **mksurfdatal** through the **mksurfdatal.pl** using input datasets that are in the **build-namelist** XML database. When you are running with input datasets that are NOT available in the XML database you either need to add them as outlined in Chapter 3, or you need to run **mksurfdatal** by hand, as we will outline here.

### Preparing your **mksurfdatal** namelist

When running **mksurfdatal** by hand you will need to prepare your own input namelist. There are sample namelists that are setup for running on the NCAR machine bluefire. You will need to change the filepaths to run on a different machine. The list of sample namelists include

```
mksurfdatal.namelist -- standard sample namelist.  
mksurfdatal.regional -- sample namelist to build for a regional grid dataset (5x5_amazon)  
mksurfdatal.singlept -- sample namelist to build for a single point grid dataset (1x1_brazil)
```

Note, that one of the inputs `mksrf_fdynuse` is a filename that includes the filepaths to other files. The filepaths in this file will have to be changed as well. You also need to make sure that the line lengths remain the same as the read is a formatted read, so the placement of the year in the file, must remain the same, even with the new filenames. One advantage of the **mksurfdatal.pl** script is that it will create the `mksrf_fdynuse` file for you.

We list the namelist items below. Most of the namelist items are filepaths to give to the input half degree resolution datasets that you will use to scale from to the resolution of your grid dataset. You must first specify the input grid dataset for the resolution to output for:

1. `mksrf_fgrid` Grid dataset

Then you must specify settings for input high resolution datafiles

1. `mksrf_ffrac` land fraction and land mask dataset
2. `mksrf_fglacier` Glacier dataset
3. `mksrf_flai` Leaf Area Index dataset
4. `mksrf_flanwat` Land water dataset
5. `mksrf_forganic` Organic soil carbon dataset
6. `mksrf_fmax` Max fractional saturated area dataset
7. `mksrf_fsoicol` Soil color dataset
8. `mksrf_fsoitex` Soil texture dataset
9. `mksrf_ftopo` Topography dataset (this is used to limit the extent of urban regions and is used for glacier multiple elevation classes)
10. `mksrf_furban` Urban dataset
11. `mksrf_fvegtyp` PFT vegetation type dataset
12. `mksrf_fvocef` Volatile Organic Compound Emission Factor dataset

You specify the ASCII text file with the land-use files.

1. `mksrf_fdynuse` "dynamic land use" for transient land-use/land-cover changes. This is an ASCII text file that lists the filepaths to files for each year and then the year it represents (note: you MUST change the filepaths inside the file when running on a machine NOT at NCAR). We always use this file, even for creating datasets of a fixed year. Also note that when using the "pft\_" settings this file will be an XML-like file with settings for PFT's rather than filepaths (see the Section called *Experimental options to mksurfdata* below).

And optionally you can specify settings for:

1. `all_urban` If entire area is urban (typically used for single-point urban datasets, that you want to be exclusively urban)
2. `mksrf_firrig` Irrigation dataset, if you want activate the irrigation model over generic cropland (experimental mode, normally NOT used)
3. `mksrf_gridnm` Name of output grid resolution (if not set the files will be named according to the number of longitudes by latitudes)
4. `mksrf_gridtype` Type of grid (default is 'global')
5. `nglcec` number of glacier multiple elevation classes. Can be 0, 1, 3, 5, or 10. When using the resulting dataset with CLM you can then run with `glc_nec` of either 0 or this value. (experimental normally use the default of 0, when running with the land-ice model in practice only 10 has been used)
6. `numpft` number of Plant Function Types (PFT) in the input vegetation `mksrf_fvegtyp` dataset.  
You change this to 20, if you want to create a dataset with prognostic crop activated. The vegetation dataset also needs to have prognostic crop types on it as well. (experimental normally not changed from the default of 16)
7. `outnc_large_files` If output should be in NetCDF large file format
8. `outnc_double` If output should be in double precision (normally we turn this on)
9. `pft_frc` array of fractions to override PFT data with for all gridpoints (experimental mode, normally NOT used).
10. `pft_idx` array of PFT indices to override PFT data with for all gridpoints (experimental mode, normally NOT used).
11. `soil_clay` percent clay soil to override all gridpoints with (experimental mode, normally NOT used).
12. `soil_color` Soil color to override all gridpoints with (experimental mode, normally NOT used).
13. `soil_fmax` Soil maximum fraction to override all gridpoints with (experimental mode, normally NOT used).
14. `soil_sand` percent sandy soil to override all gridpoints with (experimental mode, normally NOT used).

After creating your namelist, when running on a non NCAR machine you will need to get the files from the inputdata repository. In order to retrieve the files needed for mksurfdata you can do the following on your namelist to get the files from the inputdata repository, using the `check_input_data` script which also allows you to export data to your local disk.

### Example 2-7. Getting the raw datasets for mksurfdata to your local machine using the check\_input\_data script

```
> cd models/lnd/clm/tools/mksurfdata
# First remove any quotes and copy into a filename that can be read by the
# check_input_data script
> sed "s/'//g" namelist > clm.input_data_list
# Run the script with -export and give the location of your inputdata with $CSMDATA
> ../../../../../scripts/csm_utils/Tools/check_input_data -datalistdir . \
-inputdata $CSMDATA -check -export
# You must then do the same with the fdynuse file referred to in the namelist
# in this case we add a file = to the beginning of each line
> awk '{print "file = \"$1\"' pftdyn_hist_simyr2000-2000.txt > clm.input_data_list
# Run the script with -export and give the location of your inputdata with $CSMDATA
> ../../../../../scripts/csm_utils/Tools/check_input_data -datalistdir . \
-inputdata $CSMDATA -check -export
```

#### *Experimental options to mksurfdata*

The options: pft\_frc, pft\_idx, soil\_clay, soil\_color, soil\_fmax, and soil\_sand are also new and considered experimental. They provide a way to override the PFT and soil values for all grid points to the given values that you set. This is useful for running with single-point tower sites where the soil type and vegetation is known. Note that when you use pft\_frc, all other landunits will be zeroed out, and the sum of your pft\_frc array MUST equal 100.0. Also note that when using the "pft\_" options the mksrf\_fdynuse file instead of having filepaths's will be an XML-like file with PFT settings. Unlike the file of file-paths, you will have to create this file by hand, **mksurfdata.pl** will NOT be able to create it for you (other than the first year which will be set to the values entered on the command line). Note, that when PTCLM is run, it CAN create these files for you from a simpler format (see the Section called *Dynamic Land-Use Change Files for use by PTCLM* in Chapter 6). Instead of a filepath you have a list of XML elements that give information on the PFT's and harvesting for example:

```
<pft_f>100</pft_f><pft_i>1</pft_i><harv>0,0,0,0,0</harv><graz>0</graz>
```

So the **<pft\_f>** tags give the PFT fractions and the **<pft\_i>** tags give the index for that fraction. Harvest is an array of five elements, and grazing is a single value. Like the usual file each list of XML elements goes with a year, and there is limit on the number of characters that can be used.

### Standard Practices when using mksurfdata

In this section we give the recommendations for how to use **mksurfdata** to give similar results to the files that we created when using it.

If you look at the standard surface datasets that we have created and provided for use, there are three practices that we have consistently done in each (you also see these in the sample namelists and in the **mksurfdata.pl** script). The first is that we always output data in double precision (hence `outnc_double` is set to `.true.`). The next is that we always use the procedure for creating transient datasets (using `mksrf_fdynuse`) even when creating datasets for a fixed simulation year. This is to ensure that the fixed year datasets will be consistent with the transient datasets. When this is done a "surfdata.pftdyn" dataset will be created -- but will NOT be used in CLM. If you look at the sample namelist

`mksurfdata.namelist` you note that it sets `mksrf_fdynuse` to the file `pftdyn_hist_simyr2000.txt`, where the single file entered is the same PFT file used in the rest of the namelist (as `mksrf_fvegtyp`). The last practice that we always do is to always set `mksrf_ftopo`, even if glacier elevation classes are NOT active. This is important in limiting urban areas based on topographic height, and hence is important to use all the time. The glacier multiple elevation classes will be used as well if you are running a compset with the active glacier model.

There are two other important practices for creating urban single point datasets. The first is that you often will want to set `all_urban` to `.true.` so that the dataset will have 100% of the gridcell output as urban rather than some mix of: urban, vegetation types, and other landunits. The next practice is that most of our specialized urban datasets have custom values for the urban parameters, hence we do NOT want to use the global urban dataset to get urban parameters -- we use a previous version of the surface dataset for the urban parameters. However, in order to do this, we need to append onto the previous surface dataset the grid and land mask/land fraction information from the grid and fraction datasets. This is done in **mksurfdata.pl** using the NCO program **nccks**. An example of doing this for the Mexico City, Mexico urban surface dataset is as follows:

```
> nccks -A $CSMDATA/lnd/clm2/griddata/griddata_1x1pt_mexicocityMEX_c090715.nc \
$CSMDATA/lnd/clm2/surfdata/surfdata_1x1_mexicocityMEX_simyr2000_c100407.nc
> nccks -A $CSMDATA/lnd/clm2/griddata/fracdata_1x1pt_mexicocityMEX_navy_c090715.nc \
$CSMDATA/lnd/clm2/surfdata/surfdata_1x1_mexicocityMEX_simyr2000_c100407.nc
```

Note, if you look at the current single point urban surface datasets you will note that the above has already been done.

The final issue is how to build **mksurfdata**. When NOT optimized **mksurfdata** is very slow, and can take many hours to days to even run for medium resolutions such as one or two degree. So usually you will want to run it optimized. Possibly you also want to use shared memory parallelism using OpenMP with the SMP option. The problem with running optimized is that answers will be different when running optimized versus non-optimized for most compilers. So if you want answers to be the same as a previous surface dataset, you will need to run it on the same platform and optimization level. Likewise, running with or without OpenMP may also change answers (for most compilers it will NOT, however it does for the IBM compiler). However, answers should be the same regardless of the number of threads used when OpenMP is enabled. Note, that the output surface datasets will have attributes that describe whether the file was written out optimized or not, with threading or not and the number of threads used, to enable the user to more easily try to match datasets created previously. For more information on the different compiler options for the CLM4 tools see the Section called *Common environment variables and options used in building the FORTRAN tools*.

## Using NCL scripts `ndepregrid.ncl` and `aerdepregrid.ncl` to interpolate aerosol deposition datasets

Unlike the other tools, these are NCAR Command Language (NCL) scripts and you will need to get a copy of NCL in order to use them. You also won't have to build an executable in order to use them, hence no Makefile is provided. NCL is provided for free download as either binaries or source code from: <http://www.ncl.ucar.edu/>. The NCL web-site also contains documentation on NCL and its use.

By default at this point neither of these scripts *HAS* to be used, as the model is now constructed to read aerosol and Nitrogen deposition from 2-degree datasets and interpolate to the model resolution on the fly. The main reason you might want to do this now, is for better performance for single-point simulations.

Both the **ndepregrid.ncl** and **aerndepreregid.ncl** scripts have similar interfaces and you customize the output resolution and characteristics based on the settings of environment variables that you set (if you don't set any of the variables, the script has defaults that it will use). The list of environment variables that can be set are:

RES -- output resolution name  
RCP -- representative concentration pathway for future scenarios (example 2.6, 4.5, 6, or 8.5)  
SIM\_YR -- simulation year (example 1850 or 2000)  
SIM\_YR\_RNG -- simulation year range (example 1850-2000 or 1850-2100)  
GRDFIL -- full pathname of grid file to use (in place of getting the default grid file based on the RES value)  
CSMDATA -- CESM inputdata directory  
CLM\_ROOT -- root directory for CLM (models/lnd/clm directory)

**Important:** You *MUST* provide either RES or both GRDFIL *AND* RES. If you just give RES the default namelist database in `models/lnd/clm/bld` will be used to find the default grid file based on the resolution name RES. If you provide GRDFIL the input pathname of the gridfile provided will be used, and the output filename will include RES as part of its name to designate it as an output file at that resolution.

Both scripts assume that you will be interpolating from a native resolution of 1.9x2.5 and using the default files found in the namelist database to interpolate from. If you want to interpolate from another resolution or use other files, you would need to edit the scripts to do so. Both scripts also use a bilinear interpolation to do the regridding. The environment variables: RCP, SIM\_YR, and SIM\_YR\_RNG will be used to query the namelist database to determine which native dataset to interpolate from. If you don't provide valid values for these variables, it won't be able to find a dataset to interpolate from. You can use the build-namelist script to query what the valid values for these can be. Likewise, when you use RES to determine the grid file to interpolate to, it needs to be a valid value from the namelist database.

The scripts can be used to interpolate from (and create output) constant or transient datasets. Constant datasets specify the SIM\_YR and set SIM\_YR\_RNG to `constant` (which is also the default). Transient datasets need to specify both SIM\_YR and SIM\_YR\_RNG, where SIM\_YR is set to the first year in the interval (typically 1850).

The default for CSMDATA works for NCAR computers, but will need to be set to the top level directory location of your CESM input data on other computers. If you set this as a default for your shell when you login (for example with your `$HOME/.cshrc` if you use csh) you won't have to set it each time you run the script. CLM\_ROOT will default to the proper location when you run it in the `models/lnd/clm/tools/ncl_script` directory. It is only useful if you want to run the script out of a different directory.

## Using **ndepregrid.ncl** to interpolate Nitrogen deposition datasets

**ndepregrid.ncl** interpolates the Nitrogen deposition datasets from one resolution to another.

**Note:** Interpolating Nitrogen deposition files is no longer needed, because the model can read Nitrogen deposition files at one resolution and interpolate to the resolution the model is running at on the fly. Interpolating to another resolution is only useful for very coarse resolutions, if you want to save some computing resources in reading larger datasets. For example, this may be useful in obtaining single-point datasets.

For example, to interpolate to an output resolution of 0.9x1.25, for a constant simulation-year of 1850, you would do the following:

```
> env RES=0.9x1.25 SIM_YR=1850 ncl ndepregrid.ncl
```

## Using aerdepregrid.ncl to interpolate Aerosol deposition datasets

**aerdepregrid.ncl** interpolates the Aerosol deposition datasets from one resolution. It can be used to interpolate either constant datasets (for example:

`aerosoldep_monthly_2000_0.9x1.25_c090828.nc`) or transient datasets (for example:  
`aerosoldep_monthly_1849-2006_0.9x1.25_c090830.nc`).

**Note:** Interpolating aerosol deposition files is no longer needed, because the DATM model can read aerosol deposition files at one resolution and interpolate to the resolution the model is running at on the fly. Interpolating to another resolution is only useful for very coarse resolutions, if you want to save some computing resources in reading larger datasets. For example, this may be useful in obtaining single-point datasets.

For example, to interpolate to an output resolution of 4x5, for a transient simulation-year range of 1850 to 2100 and the rcp of 8.5, you would do the following:

```
> env RES=4x5 SIM_YR=1850 SIM_YR_RNG=1850-2100 RCP=8.5 ncl ndepregrid.ncl
```

## How to Customize Datasets for particular Observational Sites

There are two ways to customize datasets for a particular observational site. The first is to customize the input to the tools that create the dataset, and the second is to over-write the default data after you've created a given dataset. Depending on the tool it might be easier to do it one way or the other. In Table 3-1 we list the files that are most likely to be customized and the way they might be customized. Of those

files, the ones you are most likely to customize are: fatmlndfrc, fsurdat, faerdep (for DATM), and stream\_fldfilename\_ndep. Note **mksurfdta** as documented previously has options to overwrite the vegetation and soil types. For more information on this also see the Section called *Creating your own single-point/regional surface datasets* in Chapter 5 and PTCLM uses these methods to customize datasets see Chapter 6.

Another aspect of customizing your input datasets is customizing the input atmospheric forcing datasets. See the the Section called *Running with your own atmosphere forcing* in Chapter 5 for more information on this. Also the chapter on PTCLM in the Section called *Converting AmeriFlux Data for use by PTCLM* in Chapter 6 has information on using the AmeriFlux tower site data as atmospheric forcing.

## Conclusion of tools description

We've given a description of how to use the different tools with CLM to create customized datasets. In the next chapter we will talk about how to make these files available for build-namelist so that you can easily create simulations that include them. In the chapter on single-point and regional datasets we also give an alternative way to enter new datasets without having to edit files.

# Chapter 3. Adding New Resolutions or New Files to the build-namelist Database

In the last chapter we gave the details on how to create new files for input into CLM. These files could be either global resolutions, regional-grids or even a single grid point. If you want to easily have these files available for continued use in your development you will then want to include them in the build-namelist database so that build-namelist can easily find them for you. You can deal with them, just by editing your namelist by hand (or using a `user_nl_clm` namelist file), or by using `CLM_USRDATA_NAME`. Another way to deal with them is to enter them into the database for build-namelist, so that build-namelist can find them for you. This keeps one central database for all your files, rather than having multiple locations to keep track of files. If you have a LOT of files to keep track of it also might be easier than keeping track by hand, especially if you have to periodically update your files. If you just have a few quick experiments to try, for a short time period you might be best off using the other methods mentioned above.

There are two parts to adding files to the build-namelist database. The first part is adding new resolution names which is done in the `models/lnd/clm/bld/namelist_files/namelist_definition.xml` file (and in the `models/lnd/clm/bld/config_files/config_definition.xml` file when adding supported single-point datasets). The second part is actually adding the new filenames which is done in the `models/lnd/clm/bld/namelist_files/namelist_defaults_clm.xml` file (`models/lnd/clm/bld/namelist_files/namelist_defaults_clm_tools.xml` file for CLM tools). If you aren't adding any new resolutions, and you are just changing the files for existing resolutions, you don't need to edit the `namelist_definition` file.

## Managing Your Own Data-files

If you are running on a supported machine (such as bluefire or jaguar) the standard input datasets will already be available and you won't have to check them out of the subversion inputdata server. However, you also will NOT be able to add your own datafiles to these standard inputdata directories -- because most likely you won't have permissions to do so. In order to add files to the XML database or to use `CLM_USRDATA_NAME` you need to put data in the standard locations so that they can be found. The recommended way to do this is to use the `link_dirtree` tool in the CESM scripts. Some information on `link_dirtree` is available in the CESM1.0.4 Scripts User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>). We also have some examples of it's use here and in other sections of this User's Guide.

Using `link_dirtree` is quite simple, you give the directory where data exists and then the directory that you want to create where datasets will point to the original source files. In the example below we use `"$HOME/inputdata"`, but `MYCSMDATA` could be any directory you have access to where you want to put your data.

```
> cd scripts
# First make sure you have a inputdata location that you can write to
# You only need to do this step once, so you won't need to do this in the future
# (except to bring in any updated files in the original $CSMDATA location).
> setenv MYCSMDATA $HOME/inputdata      # Set env var for the directory for input data
> ./link_dirtree $CSMDATA $MYCSMDATA
```

Then when you create a case you will change DIN\_LOC\_ROOT\_CSMDATA to point to the location you linked to rather than the default location.

```
> ./xmlchange -file env_run.xml -id DIN_LOC_ROOT_CSMDATA -val $MYCSMDATA
```

In order to list the files that you have created you merely need to use the UNIX command **find** to find the files that are NOT softlinks. So for example executing the following command:

```
> find $MYCSMDATA -type f -print
```

for me gives the following list of CLM\_USRDAT\_NAME files that I have created.

```
/blhome/erik/inputdata/atm/cam/chem/trop_mozart_aero/aero/aerosoldep_monthly_1849-2006_1x  
/blhome/erik/inputdata/atm/cam/chem/trop_mozart_aero/aero/aerosoldep_monthly_1849-2006_13  
/blhome/erik/inputdata/atm/cam/chem/trop_mozart_aero/aero/aerosoldep_rcp8.5_monthly_1850-  
/blhome/erik/inputdata/atm/cam/chem/trop_mozart_aero/aero/aerosoldep_rcp4.5_monthly_1850-  
/blhome/erik/inputdata/atm/datm7/domain.clm/domain.lnd.1x1pt_US-Ha1_USGS.nc  
/blhome/erik/inputdata/atm/datm7/domain.clm/domain.lnd.13x12pt_f19_alaskaUSA_gx1v6.nc  
/blhome/erik/inputdata/lnd/clm2/griddata/fracdata_13x12pt_f19_alaskaUSA_gx1v6.nc  
/blhome/erik/inputdata/lnd/clm2/griddata/fracdata_1x1pt_US-Ha1_USGS.nc  
/blhome/erik/inputdata/lnd/clm2/griddata/topodata_13x12pt_f19_alaskaUSA.nc  
/blhome/erik/inputdata/lnd/clm2/griddata/griddata_1x1pt_US-Ha1.nc  
/blhome/erik/inputdata/lnd/clm2/griddata/griddata_13x12pt_f19_alaskaUSA.nc  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_13x12pt_f19_alaskaUSA_simyr1850.nc  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_1x1pt_US-Ha1_simyr2000.nc  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_pftdyn_rcp4.5_13x12pt_f19_alaskaUSA_sim  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_1x1pt_US-Ha1_simyr1850.nc  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_13x12pt_f19_alaskaUSA_simyr2000.nc  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_pftdyn_1x1pt_US-Ha1_simyr1849-2006.nc  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_pftdyn_13x12pt_f19_alaskaUSA_simyr1850-  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_pftdyn_rcp8.5_13x12pt_f19_alaskaUSA_sim  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_pftdyn_13x12pt_f19_alaskaUSA_simyr1849-  
/blhome/erik/inputdata/lnd/clm2/surfdata/surfdata_pftdyn_1x1pt_US-Ha1_simyr1850-2100.nc
```

You can also use **find** to list files that have a particular pattern in the name as well (using the **-name** option with wildcards). Also you can always rerun the **link\_dirtree** command if any new files are added that you need to be linked into your directory tree. Since, the files are soft-links -- it doesn't take up much space other than the files that you add there. This way all of the files are kept in one place, they are organized by usage according to CESM standards, and you can easily find your own files, and CLM can find them as well.

## Adding Resolution Names

If you are adding files for new resolutions which aren't covered in the namelist\_definition file -- you'll need to add them in. The list of valid resolutions is in the id="res" entry in the models/lnd/clm/bld/namelist\_files/namelist\_definition.xml file. You need to choose a name for your new resolution and simply add it to the comma delimited list of valid\_values for the id="res" entry. The convention for global Gaussian grids is number\_of\_latitudes x number\_of\_longitudes. The convention for global finite volume grids is latitude\_grid\_size x longitude\_grid\_size where latitude and longitude is measured in degrees. For regional or single-point datasets the names have a grid size number\_of\_latitudes x number\_of\_longitudes followed by an underscore and then a descriptive name such as a City name followed by an abbreviation for the Country in caps. The only hard requirement is that names be unique for different grid files. Here's what the entry for resolutions looks like in the file:

```
<entry id="res" type="char*30" category="default_settings"
      group="default_settings"
      valid_values=
      "128x256,64x128,48x96,32x64,8x16,94x192,0.23x0.31,0.47x0.63,
      0.9x1.25,1.9x2.5,2.65x3.33,4x5,10x15,5x5_amazon,1x1_tropicAtl,
      1x1_camdenNJ,1x1_vancouverCAN,1x1_mexicocityMEX,1x1_asphaltjungleNJ,
      1x1_brazil,1x1_urbanc_alpha,0.5x0.5">
  Horizontal resolutions
</entry>
```

As you can see you just add your new resolution names to the end of the valid\_values list.

When using PTCLM and adding supported single-point resolutions, you'll also want to add these resolutions to the models/lnd/clm/bld/config\_files/config\_definition.xml under the sitespf\_pt name. The entry in that file looks like:

```
<entry id="sitespf_pt"
      valid_values="none,1x1_brazil,1x1_tropicAtl,5x5_amazon,
      1x1_camdenNJ,1x1_vancouverCAN,1x1_mexicocityMEX,1x1_asphaltjungleNJ,
      1x1_urbanc_alpha,1x1_numAlA,1x1_smallvilleIA"
      value="none" category="physics">
  Flag to turn on site specific special configuration flags for supported single
  point resolutions.
  Currently the only special settings are for MEXICOCITY and VANCOUVER, which make
  changes to urban parameters.
</entry>
```

PTCLM assumes that any supported single-point resolutions are valid settings for sitespf\_pt.

## Adding or Changing Default Filenames

To add or change the default filenames you edit the models/lnd/clm/bld/namelist\_files/namelist\_defaults\_clm.xml and either change an existing filename or add a new one. Most entries in the default namelist files, include different attributes that describe the different properties that describe the differences in the datasets. Attributes include the: resolution, year to simulation, range of years to simulate for transient datafiles, the land-mask, the representative concentration pathway (rcp) for future scenarios, and the type of biogeochemistry (bgc) model used. For example the fatmgrid for the 1.9x2.5 resolution is as follows:

```
<fatmgrid hgrid="1.9x2.5" >lnd/clm2/griddata/griddata_1.9x2.5_060404.nc
</fatmgrid>
```

Other `fatmgrid` files are distinguished from this one by their resolution (hgrid) attribute.

To add or change the default filenames for CLM tools edit the  
`models/lnd/clm/bld/namelist_files/namelist_defaults_clm_tools.xml` and either  
change an existing filename or add a new one. Editing this file is similar to the  
`namelist_defaults_clm.xml` talked about above.

## What are the required files?

Different types of simulations and different types of configurations for CLM require different lists of files. The Carbon Nitrogen (cn) Biogeochemistry model for example requires `stream_fldfilename_ndep` files, which are NOT required by other bgc modes. Transient simulations also require transient datasets, and the names of these datasets are sometimes different from the static versions (sometimes both are required as in the dynamic PFT cases).

In the following table we list the different files used by CLM, they are listed in order of importance, dependencies, and customizing. So the required files are all near the top, and the files used only under different conditions are listed later, and files with the fewest dependencies are near the top, as are the files that are least likely to be customized.

**Table 3-1. Required Files for Different Configurations and Simulation Types**

Filename	Config. type	Simulation type	Resol. Dependent?	Other Dependencies
<b>Notes</b>				
fpftcon	ALL	ALL	No	No
	Not usually customized, as describes plant function type properties. PTCLM copies the file for you so that you can customize it if you like, see step 3.d.			
fsnowoptics	ALL	ALL	No	No
	Not usually customized as describes global snow optical properties.			
fsnowaging	ALL	ALL	No	No
	Not usually customized as describes global snow aging properties.			
fatmgrid	ALL	ALL	Yes	No
	Creating, using <code>mkgriddata</code> usually gives you the amount of customization you need, as it just describes the grid and grid extents.			
fatmlndfrc	ALL	ALL	Yes	land-mask

Filename	Config. type	Simulation type	Resol. Dependent?	Other Dependencies	
<b>Notes</b>		Describes the land-mask for points with active land, as well as the fraction of each grid-cell covered by land. You might customize it to make sure the land-fraction of your grid-cell matches the expected values for your site. But, usually you will just use what <code>mkgriddata</code> gives you.			
fsurdat	ALL	ALL	Yes	simulation-year	
		Describes percentages of different land-units, columns and vegetation types within each grid-cell. To customize for a specific point or region you may want to use custom input datasets to <code>mksurfdata</code> when creating the file. <code>mksurfdata</code> also allows you to customize the PFT, and soil types to it see the Section called <i>Experimental options to mksurfdata</i> in Chapter 2. PTCLM takes advantage of this to create customized datasets as well, see the chapter on PTCLM at Chapter 6.			
fpftdyn	ALL	transient land-use land-cover change	Yes	Simulation year range, and representative concentration pathway (rcp)	
		See notes on fsurdat files.			
frivinp_rtm	RTM only	ALL	No	No	
		We only provide a half-degree global river routing file. If you want to model river flow for a smaller scale, or a basin regional scale, you would need to create your own custom file to do that. Normally, we turn river-routing OFF for regional or single point simulations.			
flndtopo	ALL	fine-mesh simulations (specifying land resolution as a finer grid than atmosphere resolution).	Yes	No	
		You may customize to give better surface heights for your site, or input a higher resolution orography file when you create it using <code>mkgriddata</code> .			
fatm topo	ALL	fine-mesh simulations (specifying land resolution as a finer grid than atmosphere resolution).	Yes	No	

Filename	Config. type	Simulation type	Resol. Dependent?	Other Dependencies
	<b>Notes</b>			
	ALL	RUN_TYPE="startup", CLM_FORCE_COLDSTART="off"	Yes	mask, maxpft, bg simulation-year, start-date
	Used for starting the model from a spun-up state. Create these files by running the model for multiple years and saving the restart file from the end of a spin-up simulation.			
fglcmask	glc_nec > 0	Used for simulations with the active glacier ice sheet model "cism"	Yes	glacier-grid
	Needs to match the file used by "cism" and be for the same glacier grid. Only customized as coupled with the glacier model.			
stream_fldfilename_ndep	bgc=cn/cndv	Yes	No	simulation-year
	You may customize this file to get the Nitrogen deposition characteristics of your site if available. This file will be interpolated while the model is running from its resolution to the resolution that CLM is running at.			

# Chapter 4. How to run some special cases

In this chapter we describe how to run some special cases that take more than one step to do. The straightforward cases have compsets and/or build-namelist use-cases setup for them or require simple editing of a single-case. All of the cases here require you to do at least two simulations with different configurations, or require more complex editing of the case (changing the streams files).

The nine cases we will describe are:

1. *Running with the prognostic crop model on*
2. *Running with the irrigation model on*
3. *Spinning up the Satellite Phenology Model (CLMSP spinup)*
4. *Spinning up the biogeochemistry Carbon-Nitrogen Model (CN spinup)*
5. *Spinning up the Carbon-Nitrogen Dynamic Global Vegetation Model (CNDV spinup)*
6. *Running with MOAR data as atmospheric forcing to spinup the model*
7. *Running with your own previous simulation as atmospheric forcing to spinup the model*
8. *Doing perturbation error growth tests*
9. *Running stand-alone CLM with transient historical CO<sub>2</sub> concentration*

## Caution

The cases in this chapter are more sophisticated and require more technical knowledge and skill than cases in previous chapters. The user should be very familiar with doing simple cases before moving onto the cases described here.

## Running with the prognostic crop model on

In CLM4 in CESM1.0.3 a prognostic crop model was added to CLM4. The prognostic crop model is setup to work with CN for present day conditions and we have surface and initial condition datasets at f19 resolution. In order to use the initial condition file, we need to set the RUN\_TYPE to `startup` rather than `hybrid` since the compset for f19 sets up to use an initial condition file without crop active. To activate the crop model we simply add `"-crop on"` to CLM\_CONFIG\_OPTS.

### Example 4-1. Example Crop Simulation

```
> cd scripts
> ./create_newcase -case CROP -res f19_g16 -compset ICN -mach bluefire
> cd CROP
# Append "-crop on" to CLM_CONFIG_OPTS in env_conf.xml (you could also use an editor)
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-crop on" -append
# Change to startup type so uses spinup initial conditions file for crop if it exists
# By default the model will do a hybrid startup with an initial condition file
# incompatible with the crop surface dataset.
> ./xmlchange -file env_conf.xml -id RUN_TYPE -val startup
> ./configure -case
# Now build and run normally
```

```
> ./CROP.bluefire.build
> ./CROP.bluefire.submit
```

## Running with the irrigation model on

In CLM4 in CESM1.0.3 an irrigation model for generic crop was added to CLM4. Currently, irrigation and crop can NOT be used together see bug number 1326 in the `models/lnd/clm/doc/KnownBugs` (`..KnownBugs`) file. The irrigation model is tuned to work only with CLMSP see the caution below for more information on this. To turn on irrigation we simply add "-irrig on" to `CLM_BLDNML_OPTS`. Just as in the crop example we also change `RUN_TYPE` to `startup` so that we don't use an initial condition file that is incompatible with irrigation.

### Example 4-2. Example Irrigation Simulation

```
> cd scripts
# Note here we do a CLMSP simulation as that is what has been validated
> ./create_newcase -case IRRIG -res f19_g16 -compset I -mach bluefire
> cd IRRIG
# Append "-irrig on" to CLM_BLDNML_OPTS in env_conf.xml (you could also use an editor)
> ./xmlchange -file env_conf.xml -id CLM_BLDNML_OPTS -val "-irrig" -append
# Change to startup type so uses spinup initial conditions file for irrigation if it exists
# By default the model will do a hybrid startup with an initial condition file
# incompatible with the irrigation surface dataset.
> xmlchange -file env_conf.xml -id RUN_TYPE -val startup
> ./configure -case
# Now build and run normally
> ./IRRIG.bluefire.build
> ./IRRIG.bluefire.submit
```

### Caution

We have only run the irrigation model with CLMSP (i.e. without the CN model). We recommend that if you want to run the irrigation model with CN, that you do a spinup as outlined in the examples below. But, more than that you may need to make the adjustments we discuss in the Section called *Build-Namelist options that should NOT be exercised: in What is scientifically validated and functional in CLM4?*

## Spinning up the Satellite Phenology Model (CLMSP spinup)

To spin-up the CLMSP model you merely need to run CLMSP for 50 simulation years starting from arbitrary initial conditions. You then use the final restart file for initial conditions in other simulations. Because, this is a straight forward operation we will NOT give the details on how to do that here, but

leave it as an exercise for the reader. See the Example 4-5 as an example of doing this as the last step for CLMCN.

## Spinning up the biogeochemistry Carbon-Nitrogen Model (CN spinup)

To get the CLMCN model to a steady state, you first run it from arbitrary initial conditions using the "accelerated decomposition spinup" (-ad\_spinup in configure) mode for 600 simulation years. After this you branch from this mode in the "exit spinup" (-exit\_spinup in configure), run for a simulation year, and then save a restart from that and use it as initial conditions for further spinup of CN (at least 50 simulation years).

### Spinup of CLMCN

#### 1. AD\_SPINUP

For the first step of running 600 years in "-ad\_spinup" mode, you will setup a case, and then edit the values in `env_conf.xml` and `env_run.xml` so that the right configuration is turned on and the simulation is setup to run for the required length of simulation time. So do the following:

#### **Example 4-3. Example AD\_SPINUP Simulation**

```
> cd scripts
> ./create_newcase -case CN_spinup -res f19_g16 -compset ICN -mach bluefire
> cd CN_spinup
# Append "-ad_spinup on" to CLM_CONFIG_OPTS in env_conf.xml
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-ad_spinup on" -append
# The following sets CLM_FORCE_COLDSTART to "on" in env_conf.xml (you could also use an editor)
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
# Make the output history files only annual, by adding the following to the user_nl_clm namelist
> echo '&clm_inparm hist_nhtfrq = -8760 /' > user_nl_clm
# Now configure
> ./configure -case
> ./xmlchange -file env_run.xml -id STOP_DATE -val 6010101
# Now build
> ./CN_spinup.bluefire.build
# The following sets RESUBMIT to 30 times in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id RESUBMIT -val 30
# The following sets STOP_OPTION to "nyears" in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_OPTION -val nyears
# The following sets STOP_N to 20 years in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_N -val 20
# The following sets STOP_DATE to Jan/1 of year 601 in env_run.xml (you could also use an editor)
# Now run normally
> ./CN_spinup.bluefire.submit
```

Afterwards save the last restart file from this simulation to use in the next step.

#### 2. EXIT\_SPINIP

#### **Example 4-4. Example EXIT\_SPINUP Simulation**

```
> cd scripts
> ./create_newcase -case CN_exitspinup -res f19_g16 -compset ICN -mach bluefire
> cd CN_exitspinup
# Append "-exit_spinup on" to CLM_CONFIG_OPTS in env_conf.xml
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-exit_spinup on" -append
# Change run type to branch and branch from the last year of the last simulation
```

```

> ./xmlchange -file env_conf.xml -id RUN_TYPE      -val branch
> ./xmlchange -file env_conf.xml -id RUN_REFCASE -val CN_spinup
> ./xmlchange -file env_conf.xml -id RUN_REFDATE -val 0601-01-01
> ./xmlchange -file env_conf.xml -id GET_REFCASE -val FALSE
> ./configure -case
# Go ahead and build, so that the run directory is created
> ./CN_exitspinup.bluefire.build
# Now, Copy the last restart files from the earlier case into your run directory
> cp /ptmp/$LOGIN/archive/CN_spinup/rest/CN_spinup.*.r*.0601-01-00000* /ptmp/$LOGIN/CN_exitspinup
# And copy the rpointer files for datm and drv from the earlier case
> cp /ptmp/$LOGIN/archive/CN_spinup/rest/rpointer.atm /ptmp/$LOGIN/CN_exitspinup
> cp /ptmp/$LOGIN/archive/CN_spinup/rest/rpointer.drv /ptmp/$LOGIN/CN_exitspinup
# The following sets STOP_OPTION to "nyears" in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_OPTION -val nyears
> ./xmlchange -file env_run.xml -id STOP_N      -val 1
# Now run normally
> ./CN_exitspinup.bluefire.submit

```

### 3. Final spinup

Next save the last restart file from this step and use it as the "finidat" file to use for one more spinup for at least 50 years in normal mode. So do the following:

#### **Example 4-5. Example Final CN Spinup Simulation**

```

> cd scripts
> ./create_newcase -case CN_finalspinup -res f19_g16 -compset ICN -mach bluefire
> cd CN_finalspinup
# The following sets CLM_FORCE_COLDSTART to "on" in env_conf.xml (you could also use an editor)
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
# Now, Copy the last CLM restart file from the earlier case into your run directory
> cp /ptmp/$LOGIN/archive/CN_exitspinup/rest/CN_exitspinup.clm*.r*.0602-01-01-00000.nc \
/ptmp/$LOGIN/CN_finalspinup
# And copy the rpointer files for datm and drv from the earlier case
> cp /ptmp/$LOGIN/archive/CN_exitspinup/rest/rpointer.atm /ptmp/$LOGIN/CN_finalspinup
> cp /ptmp/$LOGIN/archive/CN_exitspinup/rest/rpointer.drv /ptmp/$LOGIN/CN_finalspinup
# Set the finidat file to the last restart file saved in previous step
> echo '&clm_inparm finidat = "CN_exitspinup.clm2.r.0602-01-01-00000.nc" /' > user_nl_clm
# Now configure
> ./configure -case
> $EDITOR Buildconf/clm.buildnml.csh
# Now build
> .CN_finalspinup.bluefire.build
# The following sets RESUBMIT to 5 times in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id RESUBMIT -val 5
# The following sets STOP_OPTION to "nyears" in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_OPTION -val nyears
# The following sets STOP_N to 10 years in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_N -val 10
# Now run as normal
> .CN_finalspinup.bluefire.submit

```

To assess if the model is spunup plot trends of CN variables of interest. If you see a trend, you may need to run the simulation longer. Finally save the restart file from the end of this simulation to use as an "finidat" file for future simulations.

# Spinning up the Carbon-Nitrogen Dynamic Global Vegetation Model (CNDV spinup)

To spinup the CLM CNDV model -- you first follow the procedures above to spinup the CN model. Then you take the CN initial state file you created for the spinup with just CN, and run CNDV for 200 more years. We've provided such spinup files for two resolutions (f09 and f19) and two time-periods (1850 and 2000), so in this example we will use the files provided to start from. We've also provided a spinup file at f19 resolution for CNDV, hence the following is NOT required when running at f19. If you were to start from your own CLMCN spinup files -- the procedure would require some modification. There are no compsets using CNDV, so in `env_conf.xml` change `CLM_CONFIG_OPTS` to `-bgc cndv`.

## Example 4-6. Example CNDV Spinup Simulation

```
> cd scripts
> ./create_newcase -case CNDV_spinup -res f09_g16 -compset ICN -mach bluefire
> cd CNDV_spinup
# Set run type to startup and do a cold start
> ./xmlchange -file env_conf.xml -id RUN_TYPE -val startup
# The following sets CLM_CONFIG_OPTS to "-bgc cndv" in env_conf.xml (you could also use an editor)
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-bgc cndv"
# Make the default primary history file annual and add an annual 1D vector auxiliary file
# By putting the following in a user_nl_clm file.
> cat << EOF > user_nl_clm
&clm_inparm
hist_nhtfrq = -8760, -8760
hist_mfilt = 1, 1
hist_finc12 = 'TLAI', 'TSAI', 'HTOP', 'HBOT', 'NPP'
hist_dov2xy = .true., .false.
/
> ./configure -case
# NOTE: If you were using your own CN spinup files you would edit the namelist to use it
# $EDITOR Buildconf/clm.buildnml.csh
#
# Now build and run as normal
> ./CNDV_spinup.bluefire.build
# The following sets RESUBMIT to 10 times in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id RESUBMIT -val 10
# The following sets STOP_OPTION to "nyears" in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_OPTION -val nyyears
# The following sets STOP_N to 20 years in env_run.xml (you could also use an editor)
> ./xmlchange -file env_run.xml -id STOP_N -val 20
# Make sure you turn archiving on, so you save your files to long term archival
> ./xmlchange -file env_run.xml -id DOUT_L_MS -val TRUE
> ./CNDV_spinup.bluefire.submit
```

In a data analysis tool you should examine the auxiliary file and examine the `pftsld_wtgcell` to see where and what types of vegetation have been established. See the caution in Example 1-7 for more information on visualizing and analyzing 1D vector fields.

**Note:** CNDV also writes out two vector fields to "hv" auxiliary files, on an annual basis by default.

**Note:** We've provided a spinup file for CNDV at f19 resolution, you could also use `interpinic` to interpolate this file to other resolutions.

## Running with MOAR data as atmospheric forcing to spinup the model

Because it takes so long to spinup the CN model (as we just saw previously), if you are doing fully coupled simulations with active atmosphere and ocean, you will want to do the spinup portion of this "offline". So instead of doing expensive fully coupled simulations for the spinup duration, you run CLM in a very cheap "I" compset using atmospheric forcing from a shorter fully coupled simulation (or a simulation run previously by someone else).

In this example we will use the `I1850SPINUPCN` compset to setup CLM to run with atmospheric forcing from a previous fully coupled simulation with data that is already stored on disk on bluefire. There are several simulations that have high frequency data for which we can do this. You can also do this on a machine other than bluefire, but would need to download the data from the Earth System Grid and change the datapath similar to Example 4-9.

### Example 4-7. Example Simulation with MOAR Data on bluefire

```
> cd scripts
> ./create_newcase -case MOARforce1850 -res f19_g16 -compset I1850SPINUPCN -mach bluefire
> cd MOARforce1850
# The following sets the casename to point to for atm forcing (you could also use an editor)
> ./xmlchange -file env_conf.xml -id DATM_CPL_CASE -val b40.1850.track1.1deg.006a
# The following sets the align year and years to run over for atm forcing
# (you could also use an editor)
> ./xmlchange -file env_conf.xml -id DATM_CPL_YR_ALIGN -val 1
> ./xmlchange -file env_conf.xml -id DATM_CPL_YR_START -val 960
> ./xmlchange -file env_conf.xml -id DATM_CPL_YR_END -val 1030
> ./configure -case
# Now build and run as normal
> ./MOARforce1850.bluefire.build
> ./MOARforce1850.bluefire.submit
```

### Caution

Because of bug 1339 (see the `models/lnd/clm/doc/KnownBugs` (`..//KnownBugs`) file on this) you can't run with 83 or more years of forcing. If you do need to run with more years of forcing, you'll need to address the issue as outlined in the `models/lnd/clm/doc/KnownBugs` (`..//KnownBugs`) file.

## Running with your own previous simulation as atmospheric forcing to spinup the model

Another way that you might want to spinup the model is to run your own simulation for a relatively short period (either a B, E, or F compset) and then use it as forcing for your "I" case later. By only running 20 to 50 years for the fully coupled case, you'll save a substantial amount of computer time rather than running the entire spinup period with a fully coupled model.

The first thing we need to do is to run a fully coupled case and save the atmospheric coupling fields on a three hourly basis. In this example, we will run on bluefire and archive the data to a local disk that we can then use in the next simulation.

### Example 4-8. Example Fully Coupled Simulation to Create Data to Force Next Example Simulation

```
> cd scripts
> ./create_newcase -case myBCN1850 -res f09_g16 -compset B1850CN -mach bluefire
> cd myBCN1850
> ./configure -case
# Set histaux_a2x3hr to .true. in cpl.buildnml.csh so output from the atmosphere model
# will be saved 3 hourly
$EDITOR BuildConf/cpl.buildnml.csh
# Now build
> ./myBCN1850.bluefire.build
# The following sets the archival disk space (you could also use an editor)
> ./xmlchange -file env_run.xml -id DOUT_S_ROOT -val '/glade/home/$USER/$CASE'
# Make sure files are archived to disk, but NOT to long term storage
# (you could also use an editor)
> ./xmlchange -file env_run.xml -id DOUT_S -val TRUE
> ./xmlchange -file env_run.xml -id DOUT_L_MS -val FALSE
# Set the run length to run a total of 20 years (you could also use an editor)
> ./xmlchange -file env_run.xml -id RESUBMIT -val 9
> ./xmlchange -file env_run.xml -id STOP_OPTION -val nyears
> ./xmlchange -file env_run.xml -id STOP_N -val 2
# Now run as normal
> ./myBCN1850.bluefire.submit
```

Now we run an I compset forced with the data from the previous simulation using the CPLHIST3HrWx option to DATM\_MODE. See the Section called *CPLHIST3HrWx mode and it's DATM settings* in Chapter 1 for more information on the DATM settings for CPLHIST3HrWx mode.

### Example 4-9. Example Simulation Forced with Data from the Previous Simulation

```
> cd scripts
> ./create_newcase -case frcwmyBCN1850 -res f09_g16 -compset I1850SPINUPCN -mach bluefire
> cd frcwmyBCN1850
# The following sets the casename to point to for atm forcing (you could also use an editor)
> ./xmlchange -file env_conf.xml -id DATM_CPL_CASE -val "myBCN1850"
# The following sets the align year and years to run over for atm forcing
# (you could also use an editor)
> ./xmlchange -file env_conf.xml -id DATM_CPL_YR_ALIGN -val "1"
> ./xmlchange -file env_conf.xml -id DATM_CPL_YR_START -val "1"
> ./xmlchange -file env_conf.xml -id DATM_CPL_YR_END -val "20"
# Set the datapath in the template to the archival path from the case above
> sed -E 's#set datapath = ".+"#set datapath = "/glade/home/$USER/%c/cpl/hist"#' \
Tools/Templates/datm.cpl7.template > new.datm.cpl7.template
> mv -f new.datm.cpl7.template Tools/Templates/datm.cpl7.template
> chmod +x Tools/Templates/datm.cpl7.template
> ./configure -case
# Now build and run as normal
> ./frcwmyBCN1850.bluefire.build
> ./frcwmyBCN1850.bluefire.submit
```

**Note:** In order to accomplish this we needed to edit the DATM template file. See Appendix A for more information on doing this. If your input case was at a resolution besides f09 you would have to edit the DATM template file even further to use a domain file at the input resolution.

## Doing perturbation error growth tests

Doing perturbation error growth tests is a way to validate a port of the model to a new machine or to verify that changes are only roundoff. The steps are the same in either case, but in the discussion below I will assume you are doing a port validation to a new machine (but in parentheses I will put a reminder that it could also be for code-mods). The basic idea is to run a case on the trusted machine (trusted code) and another with initial conditions perturbed by roundoff and compare the results of the two. The difference between these two simulations (the error) will grow over time and describe a curve that we compare with the error growth on the new machine (code changes). The error growth on the new machine is the difference between the non-perturbed state on the trusted machine and the non-perturbed state on the new machine (code changes). If the new machine (code changes) are well-behaved the plot of this error growth compared to the error growth curve on the trusted machine should be similar. If the changes are NOT well-behaved the changes from the new machine (code changes) will be larger than the perturbation changes. In summary the simulations and steps that need to be performed are:

1. Run a simulation with the trusted code on the trusted machine. (optionally you can use a dataset from inputdata repository).
2. Run a simulation with the trusted code on the trusted machine with initial conditions perturbed by roundoff (using a namelist item to do so). (this is optional if you are using inputdata repository datasets)
3. Run a simulation with the new code on the non-trusted machine (code changes).
4. Do a plot of the RMS difference of history variables between simulation 1 and simulation 2.
5. Do a plot of the RMS difference of history variables between simulation 1 and simulation 3.
6. Compare the two plots in steps 4 and 5.
7. If the plots compare well the new machine (code changes) is running as well as the trusted machine.
8. If the plots do *NOT* compare well the new machine is *NOT* running as well as the trusted machine.

Typically the recommendation here is to lower the optimization level on the new machine and try again (or in the case of code changes, modify or simplify the code changes to get something that should be closer).

The history variables we have used to do this is either 'TSOI', and/or 'TSA'. 'TSOI' are the 3D snow and soil temperatures for vegetated land-units. If there is a change in soil physics it should show up in this field (and it should show up even for something that is at a pretty deep soil depth). However, as 'TSOI' is only for vegetated land-units, changes in lake or urban land-units -- will NOT show up. 'TSA' by contrast is the 2m surface temperature across all land-units, so changes in urban or lake land-units will show up. However, changes in deep soil physics will only show up as it propagates to the surface. So one field may show something that the other doesn't. In the examples, we use 'TSOI', but 'TSA' can be used as well. And in most cases you should check both.

Now we will give a detailed description of the procedure with examples and the exact steps to perform.

## Using Perturbation Error Growth Analysis to Verify a Port to a New Machine

### 1. Running non-perturbed on trusted machine

The first step is to run a non-perturbed case on the trusted machine. You need to run all of the steps with the same compset and same resolution. For these examples we will use 2-degree resolution with the ICN compset for 2000 conditions. You need to run for three days with a cold-start.

**Note:** As we describe below, This is optional if you will be using datasets from the inputdata repository to take place of this step.

### Example 4-10. Example non-perturbed error growth simulation

```
> cd scripts
> ./create_newcase -case trustedMachinePergro0 -compset ICN -res f19_g16 \
-mach bluefire
> cd trustedMachinePergro0
# Set the non-perturbed PERGRO use-case
> ./xmlchange -file env_conf.xml -id CLM_NML_USE_CASE -val pergro0_pd
# Set coldstart on so arbitrary initial conditions will be used
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
> ./xmlchange -file env_conf.xml -id RUN_TYPE -val startup
# Set PERGRO on in the configure
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-pergro on" -append
# Now configure and build
> ./configure -case
> ./trustedMachinePergro0.bluefire.build
# Set it to run for three days and turn archiving off
> ./xmlchange -file env_run.xml -id STOP_N -val 3
> ./xmlchange -file env_run.xml -id DOUT_S -val FALSE
# Run the case and then you will save the history file output for later use
> ./trustedMachinePergro0.bluefire.submit
```

**Note:** If you aren't able to do this step, as you don't have access to a trusted machine, you can use datasets that are available from the svn inputdata repository to take place of running it yourself. The disadvantage is that this is only done for certain model versions and for exactly the configuration/namelist given here. You won't be able to test it for your own custom code or configurations.

### 2. Running perturbed on the trusted machine

The next step is to run a perturbed case on the trusted machine.

### Example 4-11. Example perturbed error growth simulation

```
> cd scripts
> ./create_newcase -case trustedMachinePergroRnd -compset ICN -res f19_g16 \
-mach bluefire
> cd trustedMachinePergroRnd
# Set the perturbed PERGRO use-case
> ./xmlchange -file env_conf.xml -id CLM_NML_USE_CASE -val pergro_pd
# Set coldstart on so arbitrary initial conditions will be used
> ./xmlchange -file env_conf.xml -id RUN_TYPE -val startup
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
# Set PERGRO on in the configure
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-pergro on" -append
```

```

# Now configure and build
> ./configure -case
> ./trustedMachinePergroRnd.bluefire.build
# Set it to run for three days and turn archiving off
> ./xmlchange -file env_run.xml -id STOP_N -val 3
> ./xmlchange -file env_run.xml -id DOUT_S -val FALSE
# Run the case and then you will save the history file output for later use
> ./trustedMachinePergroRnd.bluefire.submit

```

**Note:** If you aren't able to do this step, as you don't have access to a trusted machine, you can use datasets that are available from the svn inputdata repository to take place of running it yourself. The disadvantage is that this is only done for certain model versions and for exactly the configuration/namelist given here. You won't be able to test it for your own custom code or configurations.

### 3. Running non-perturbed on the new machine

The next step is to run a non-perturbed case on the new machine. Here we will demonstrate using the machine intrepid. For the previous two steps you have the option of using datasets provided in the subversion inputdata repository to take their place -- however this step is required.

```

> cd scripts
> ./create_newcase -case newMachinePergro0 -compset ICN -res f19_g16 \
-mach intrepid
> cd newMachinePergro0
# Set the non-perturbed PERGRO use-case
> ./xmlchange -file env_conf.xml -id CLM_NML_USE_CASE -val pergro0_pd
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
> ./xmlchange -file env_conf.xml -id RUN_TYPE -val startup
# Set PERGRO on in the configure
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-pergro on" -append
# Now configure and build
> ./configure -case
> ./newMachinePergro0.intrepid.build
# Set it to run for three days and turn archiving off
> ./xmlchange -file env_run.xml -id STOP_N -val 3
> ./xmlchange -file env_run.xml -id DOUT_S -val FALSE
# Run the case and then you will save the history file output for later use
> ./newMachinePergro0.intrepid.submit

```

### 4. Plotting the differences

You can use the **cprnc** program to compute root mean square differences between the relevant history files. See the Section called *Using the cprnc tool to compare two history files* in Chapter 2 for more information on it and how to build it. On many platforms you will need to set some environment variables in order to complete the build (see the Section called *Common environment variables and options used in building the FORTRAN tools* in Chapter 2 for more information on building the tools).

```

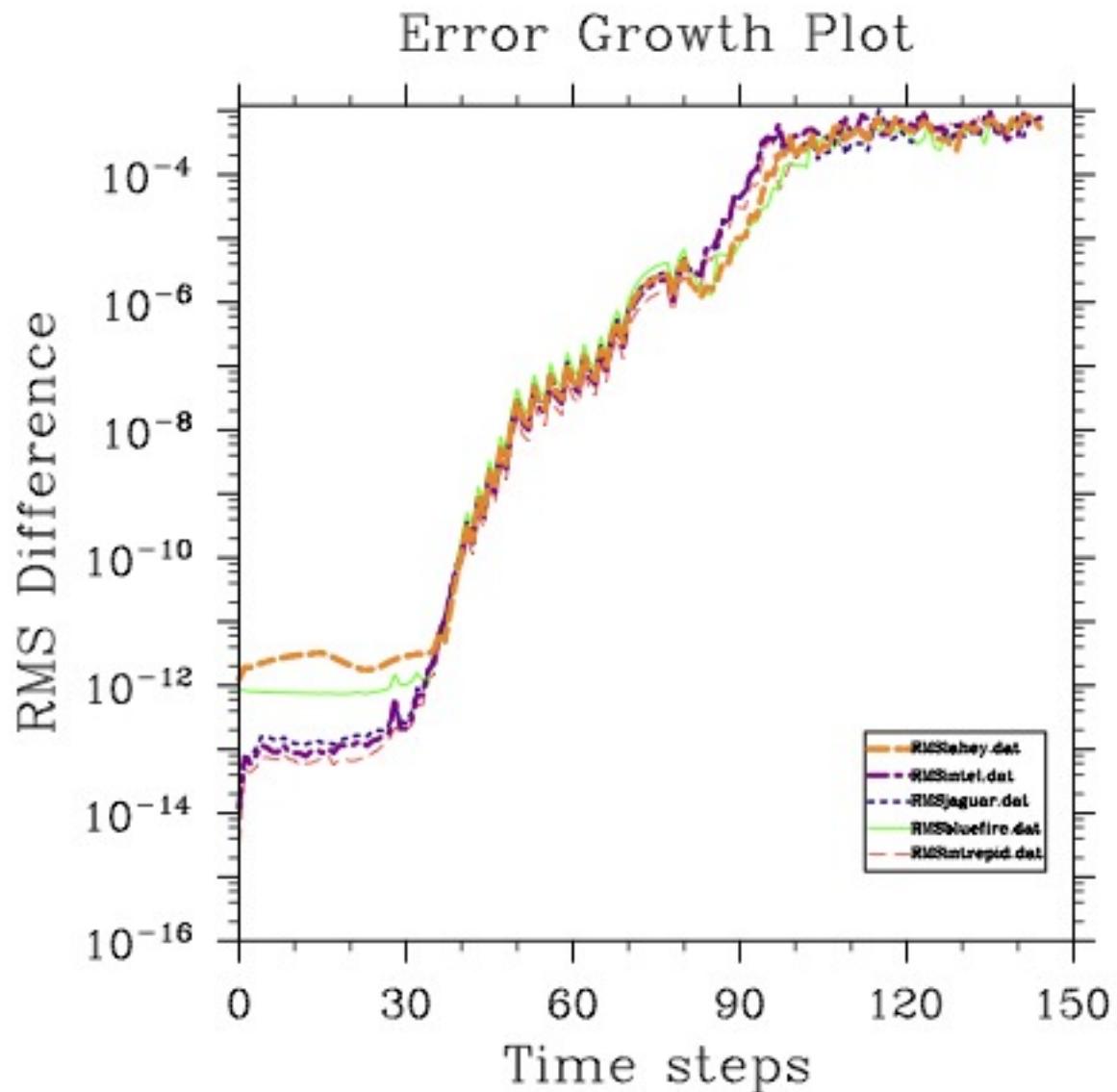
# Build the cprnc program
> cd models/lnd/clm/tools/cprnc
> gmake
# Now go to your case directory and run cprnc on the trusted-machine with and without
# perturbation
> cd ../../../../scripts/trustedMachinePergro0
> ../../models/lnd/clm/tools/cprnc/cprnc trustedMachinePergro0.clm2.h0.001-01-01.00000.nc \
..../trustedMachinePergroRnd/trustedMachinePergroRnd.clm2.h0.001-01-01.00000.nc > trustedPergro.log
# Copy the history file from the new machine to here
#
# And now run cprnc on the trusted-machine and the new machine both without perturbation
> ../../models/lnd/clm/tools/cprnc/cprnc trustedMachinePergro0.clm2.h0.001-01-01.00000.nc \
..../newMachinePergro0/newMachinePergro0.clm2.h0.001-01-01.00000.nc > newPergro.log

```

```
# Now extract out the RMS differences of TSOI for both
# You may want to extract out the RMS differences for TSA as well
# Changes in urban or lake land-units won't be detected with TSOI
> grep "RMS TSOI" trustedPergro.log | awk '{print $3}' > RMStrusted.dat
> grep "RMS TSOI" newPergro.log      | awk '{print $3}' > RMSnewmachine.dat
# And plot the two curves up to your screen
> env TYPE=x11 RMSDAT=RMSnewmachine.dat RMSDAT2=RMStrusted.dat ncl \
.../models/lnd/clm/tools/ncl_scripts/pergroPlot.ncl
```

Here is a sample plot for several trusted machines: bluefire, intrepid, jaguar, and edinburgh (with both the lahey and intel compilers). The green line is the error growth for bluefire, the red is the error growth for intrepid, the dashed navy is for jaguar, the dashed maroon is for the intel compiler on edinburgh, and the thick dashed goldenrod line is for edinburgh with the lahey compiler. Note, the data for this plot is in `models/lnd/clm/tools/ncl_scripts` the files are named: according to the legend. Note, that the lines tend to cluster together and follow quite closely to the bluefire line which is our main trusted machine.

Figure 4-1. Sample Good Perturbation Error Growth Curves (within roundoff)



When you do NOT have access to a trusted machine you can use the trusted file from bluefire that is available on the inputdata repository.

```
# Build the cprnc program
> cd models/lnd/clm/tools/cprnc
> gmake
# Get the unperturbed file from the subversion repository
> cd ../../../../scripts/newMachinePergro0
> set dir = "lnd/clm2/pergrodata"
> set file = bluefirePergro0.ICN.0001-01-01_1.9x2.5_gx1v6_simyr2000_clm4-cesml_0_3.cl10617.nc
```

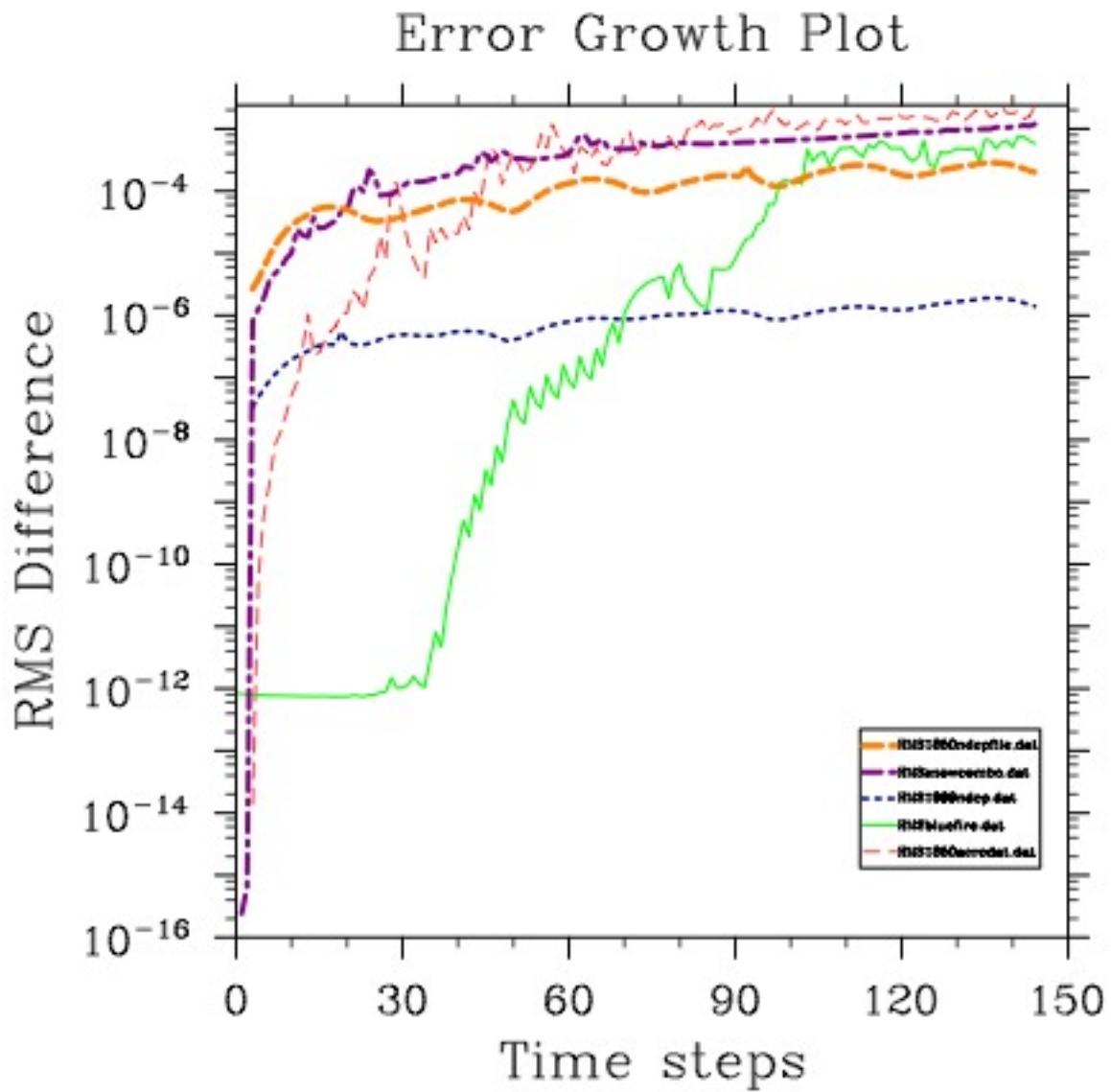
```

> echo "trustedfile = DIN_LOC_ROOT/$dir/$file" > clm.input_data_list
> ../../ccsm_utils/Tools/check_input_data -datalistdir . -export -inputdata $DIN_LOC_ROOT
# And now run cprnc on the bluefire file and the new machine both without perturbation
> ../../models/lnd/clm/tools/cprnc/cprnc $file \
..../newMachinePergro0/newMachinePergro0.clm2.h0.001-01-01.00000.nc > newPergro.log
# Now extract out the RMS difference
# You may want to extract out the RMS differences for TSA as well
# Changes in urban or lake land-units won't be detected with TSOI
> grep "RMS TSOI" newPergro.log | awk '{print $3}' > RMSnewmachine.dat
# And plot the new curve versus the trusted curve up to your screen
> env TYPE=x11 RMSDAT=RMSnewmachine.dat \
RMSDAT2=../../models/lnd/clm/tools/ncl_scripts/RMSbluefire.dat \
..../models/lnd/clm/tools/ncl_scripts/pergroPlot.ncl

```

In the figure below we now show example of curves for changes that are larger than roundoff. Once again the green curve is the trusted error growth from bluefire. The other curves are for changes that may be fairly small, but are larger than roundoff. The goldenrod curve is for using the 1850, and the navy is for using the 1999 Nitrogen deposition files rather than for year 2000. The red is for using the 1850 aerosol dataset rather than 2000, and the maroon is for adding the snow combination bug in. The differences in changes that are greater than roundoff is that the curves climb very steeply to the  $10^{-6}$  value and then level off, while the curve for bluefire climbs much more slowly and gradually. The curves also don't mimic each other in any way, like the trusted machine plots do.

Figure 4-2. Sample Bad Perturbation Error Growth Curves (changes greater than roundoff)



## Running stand-alone CLM with transient historical CO<sub>2</sub> concentration

In this case you want to run a simulation with stand-alone CLM responding to changes in CO<sub>2</sub> for a historical period. For this example, we will start with the "I\_1850-2000\_CN" compset that has transient:

land-use, Nitrogen and Aerosol deposition already. You could also use another compset if you didn't want these other features to be transient. In order to get CO<sub>2</sub> to be transient we need to edit the DATM template so that we add an extra streams file to describe how CO<sub>2</sub> varies over the historical period. You also need a NetCDF datafile that datm can read that gives the variation. You could supply your own file, but we have a standard file that is used by CAM for this and our example will make use of this file.

**Note:** Most everything here has to do with changing datm rather than CLM to allow this to happen. As such the user that wishes to do this should first become more familiar with datm and read the CESM Data Model User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/data8>) especially as it pertains to the datm. Note, also that in this example we show how to edit the datm "buildnml" file for your case, but you could do something similar by editing the datm template.

## Warning

This section documents the process for doing something that is non-standard. There may be errors with the documentation and process, and you may have to do some work before all of this works for you. If that is the case, we recommend that you do further research into understanding the process and the files, as well as understanding the datm and how it works. You may have to read documentation found in the code for datm as well as "csm\_share".

The datm has "streams" files that have rough XML-like syntax and specify the location and file to get data from, as well as information on the variable names and the data locations of the grid points. The datm expects specific variable names and the datm "maps" the expected variable names from the file to the names expected by datm. The file we are working with here is a file with a single-point, that covers the entire globe (so the vertices go from -90 to 90 degrees in latitude and 0 to 360 degrees in longitude). Since it's a single point it's a little easier to work with than datasets that may be at a given horizontal resolution. The datm also expects that variables will be in certain units, and only expects a limited number of variables so arbitrary fields can NOT be exchanged this way. However, the process would be similar for datasets that do contain more than one point.

The three things that are needed: a domain file, a data file, and a streams text file. The domain file is a CF-compliant NetCDF file that has information on the grid points (latitudes and longitudes for cell-centers and vertices, mask, fraction, and areas). The datafile is a CF-compliant NetCDF file with the data that will be mapped. The streams text file is the XML-like file that tells datm how to find the files and how to map the variables datm knows about to the variable names on the NetCDF files. Note, that in our case the domain file and the data file are the same file. In other cases, the domain file may be separate from the data file.

First we are going to create a case, and we will edit the `Buildconf/datm.buildnml.csh` so that we add a CO<sub>2</sub> data stream in. There is a streams text file available in `models/lnd/clm/doc/UsersGuide/co2_streams.txt`, that includes file with a CO<sub>2</sub> time-series from 1765 to 2007.

### Example 4-12. Example Transient Simulation with Historical CO<sub>2</sub>

```
> cd scripts
> ./create_newcase -case DATM_CO2_TSERIES -res f19_g16 -compset I_1850-2000_CN \
-mach bluefire
```

```
> cd DATM_CO2_TSERIES
# Set CCSM_BGC to CO2A so that CO2 will be passed from atmosphere to land
> ./xmlchange -file env_conf.xml -id CCSM_BGC -val CO2A
# Set CLM_CO2_TYPE to diagnostic so that the land will use the value sent from the atmosphere
> ./xmlchange -file env_conf.xml -id CLM_CO2_TYPE -val diagnostic
> ./configure -case
> cd Buildconf
# Copy the sample streams file over
> cp ../../models/lnd/clm/doc/UsersGuide/co2_streams.txt .
```

The first thing we will do is to edit the datm buildnml script to add a CO2 file stream in. To do this we will apply a patch with the differences needed. The patch file `addco2_datm.buildnml.diff` is in `models/lnd/clm/doc/UsersGuide` and looks like this...

```
*** datm.buildnml.csh.orig 2010-06-11 10:59:29.246523532 -0600
--- datm.buildnml.csh 2010-06-11 11:06:30.710784206 -0600
*****
*** 34,48 ****
    streams      = 'clm_qian.T62.stream.Solar.txt 1895 1948 1972 ',
                  'clm_qian.T62.stream.Precip.txt 1895 1948 1972 ',
                  'clm_qian.T62.stream.TPQW.txt 1895 1948 1972 ',
!                 'presaero.stream.txt 1849 1849 2006'
    vectors      = 'null'
    mapmask      = 'nomask',
                  'nomask',
                  'nomask',
                  'nomask'
    tintalgo     = 'coszen',
                  'nearest',
                  'linear',
                  'linear'
/
EOF1
--- 34,56 ----
    streams      = 'clm_qian.T62.stream.Solar.txt 1895 1948 1972 ',
                  'clm_qian.T62.stream.Precip.txt 1895 1948 1972 ',
                  'clm_qian.T62.stream.TPQW.txt 1895 1948 1972 ',
!                 'presaero.stream.txt 1849 1849 2006',
!                 'datm.global1val.stream.CO2.txt 1766 1766 2005 '
    vectors      = 'null'
    mapmask      = 'nomask',
                  'nomask',
                  'nomask',
                  'nomask'
+
    mapalgo      = 'bilinear',
                  'bilinear',
                  'bilinear',
                  'bilinear',
                  'nn'
    tintalgo     = 'coszen',
                  'nearest',
                  'linear',
+
                  'linear',
                  'linear'
/
EOF1
*****
*** 1112,1121 ***
--- 1120,1132 ----
</streamstemplate>
EOF1

+ cp $CASEBUILD/co2_streams.txt datm.global1val.stream.CO2.txt
+
$CASETOOLS/listfilesin_streams -input_data_list -t clm_qian.T62.stream.Solar.txt >> $CASEBUILD/datm.input_data_list
$CASETOOLS/listfilesin_streams -input_data_list -t clm_qian.T62.stream.Precip.txt >> $CASEBUILD/datm.input_data_list
$CASETOOLS/listfilesin_streams -input_data_list -t clm_qian.T62.stream.TPQW.txt >> $CASEBUILD/datm.input_data_list
+ $CASETOOLS/listfilesin_streams -input_data_list -t datm.global1val.stream.CO2.txt >> $CASEBUILD/datm.input_data_list

cat >! presaero.stream.txt << EOF1
<streamstemplate>
```

So to apply the patch you do this...

```
> cd scripts/DATM_CO2_TSERIES/Buildconf  
> patch < ../../models/lnd/clm/doc/UsersGuide/addco2_datm.buildnml.diff
```

Once, you've done that you can build and run your case normally.

## Warning

The patch assumes you are using a `I_1850-2000_CN` compset out of the box, with `DATM_PRESAERO` equal to `trans_1850-2000`. So it assumes standard Qian atmosphere forcing, and transient prescribed aerosols from streams files. If your case changes anything here the patch will fail, and you will need to put the changes in by hand.

**Note:** If the patch fails, you will have to add the changes to the `datm.buildnml.csh` found in the above patch file by hand. Basically, it adds an extra streams file for  $\text{CO}_2$  to the end of the streams variable, and other arrays associated with streams (adding mapalgo as a new array with bilinear for everything, but the  $\text{CO}_2$  file which should be "nn" for nearest neighbor).

## Warning

The streams file above is hard-coded for the path of the file on NCAR computers. To use it on an outside machine you'll need to edit the filepath in the streams file to point to the location where you have the file.

After going through these steps, you will have a case where you have datm reading in an extra streams text file that points to a data file with  $\text{CO}_2$  data on it that will send that data to the CLM.

# Chapter 5. How to run Single-Point/Regional cases

The CLM also allows you to set up and run cases with a single-point or a local region as well as global resolutions. This is often useful for running quick cases for testing, evaluating specific vegetation types, or land-units, or running with observed data for a specific site. There are four different ways to do this: `PTS_MODE`, `CLM_PT1_NAME`, `CLM_USRDATA_NAME`, and with `PTCLM`.

`PTS_MODE` -- to run for a single point using global datasets.

`CLM_PT1_NAME` -- to run for a supported single-point or regional dataset.

`CLM_USRDATA_NAME` -- to run using your own datasets (single-point or regional).

`PTCLM` -- to easily setup simulations to run for tower sites..

**Note:** `PTS_MODE` and `PTCLM` only work for a single point, while the other two options can also work for regional datasets as well.

## Which Single Point Option Should I choose?

In general the Section called *Running PTS\_MODE configurations* is the quick and dirty method that gets you started without having to create datasets -- but has limitations. It's good for an initial attempt at seeing results for a point of interest, but since you can NOT restart with it, its usage is limited. It is the quickest method as you can create a case for it directly from `create_newcase`. Although you can't restart, running a single point is very fast, and you can run for long simulation times even without restarts. If you need restarts a good solution is to use `getregional_datasets.pl` and `CLM_USRDATA_NAME` which can get you running almost as quickly as well as `PTS_MODE`. Like `PTS_MODE` the Section called *Using getregional\_datasets.pl to get a complete suite of single-point/regional surface datasets from global ones* only runs for points that exist within a global dataset.

Running `CLM_PT1_NAME` is a great solution, if one of the supported single-point/regional datasets, is your region of interest (see the Section called *Running Supported Single-point/Regional Datasets*). All the datasets are created for you, and you can easily select one and run, pretty much, out of the box with it. The problem is that there is a very limited set of supported datasets. You can also use this method for your own datasets, but you have to create the datasets, and add them to the XML database and to the DATM. This is worthwhile if you want to repeat many multiple cases for a given point or region.

Next, `CLM_USRDATA_NAME` is the best way to setup cases quickly where you have to create your own datasets (see the Section called *Creating your own single-point/regional surface datasets*). With this method you don't have to change DATM or add files to the XML database -- but you have to follow a strict naming convention for files. However, once the files are named and in the proper location, you can easily setup new cases that use these datasets. This is good for treating all the required datasets as a "group" and for a particular model version. For advanced CLM developers who need to track dataset changes with different model versions you would be best off adding these datasets as supported datasets with the `CLM_PT1_NAME` method.

Lastly *PTCLM* is a great way to easily create datasets, setup simulations and run simulations for tower sites. It takes advantage of both *CLM\_PT1\_NAME* and *CLM\_USRDAT\_NAME* internally. A big advantage to it, is that it's one-stop shopping, it runs tools to create datasets, and runs **create\_newcase** and sets the appropriate env variables for you. So you only have to learn how to run one tool, rather than work with many different ones. *PTCLM* is described in the next chapter Chapter 6.

Finally, if you also have meteorology data that you want to force your CLM simulations with you'll need to setup cases as described in the Section called *Running with your own atmosphere forcing*. You'll need to create CLM datasets either according to *CLM\_PT1\_NAME* or *CLM\_USRDAT\_NAME*, but you'll also need to modify DATM to use your forcing data. And you'll need to change your forcing data to be in a format that DATM can use. In the *PTCLM* chapter the the Section called *Converting AmeriFlux Data for use by PTCLM* in Chapter 6 section tells you how to use AmeriFlux data for atmospheric forcing.

## Running PTS\_MODE configurations

*PTS\_MODE* enables you to run the model using global datasets, but just picking a single point from those datasets and operating on it. It can be a very quick way to do fast simulations and get a quick turnaround.

To setup a *PTS\_MODE* simulation you use the "-pts\_lat" and "-pts\_lon" arguments to **create\_newcase** to give the latitude and longitude of the point you want to simulate for (the code will pick the point on the global grid nearest to the point you give. Here's an example to setup a simulation for the nearest point at 2-degree resolution to Boulder Colorado.

```
> cd scripts
> ./create_newcase -case testPTS_MODE -res f19_g16 -compset I -mach bluefire \
-pts_lat 40.0 -pts_lon -105
> cd testPTS_MODE
# We make sure the model will start up cold rather than using initial conditions
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
> ./xmlchange -file env_conf.xml -id RUN_TYPE -val startup
```

Then configure, build and run as normal. We make sure initial conditions are NOT used since *PTS\_MODE* currently CAN NOT run with initial conditions.

**Important:** By default it sets up to run with *USE\_MPISERIAL* (in the *env\_build.xml* file) turned on, which allows you to run the model interactively. On some machines this mode is NOT supported and you may need to change it to FALSE before you are able to build.

### Warning

*PTS\_MODE* currently does *NOT* restart nor is it able to startup from global initial condition files. See bugs "1017 and 1025" in the *models/lnd/clm/doc/KnownLimitations* (*..//KnownLimitations*) file.

**Note:** You can change the point you are simulating for at run-time by changing the values of *PTS\_LAT* and *PTS\_LON* in the *env\_run.xml* file.

**Note:** Note, that when running with PTS\_MODE the number of processors is automatically set to one. When running a single grid point you can only use a single processor. You might also want to set the "env\_conf" variable: USE\_MPISERIAL to TRUE so that you can also run interactively without having to use MPI to start up your job.

## Warning about Running with a Single-Processor on a Batch Machine

This problem always comes up when running for a single point, because you can only use a single-processor, but may come up in other instances when you are running with one processor. This applies to all the different ways of running in single-point mode.

### Warning

A warning for submitting single-point simulations to the batch que when only using one processor. On many machines this will mean using up at least an entire node, and being charged for all the CPU's on that node even if you aren't using them. For example, on the NCAR machine bluefire, there are 32 processors for each node and the batch scripts are setup to have exclusive use of that node (and hence be charged for all 32 processors). There are similar issues on other machines, below we show you what to do when running on bluefire.

To change this on bluefire -- change the following:

```
#BSUB -q regular
#BSUB -N
#BSUB -x
```

to...

```
#BSUB -q share
#BSUB -N
```

so remove the "#BSUB -x" which gives you the entire node exclusively, and change to the share que. One other machines you may have to do something similar, but the particulars depend on the given machine, hence you will need to consult with the system administrators for the given machine you are running on.

**Note:** Another similar problem on many machines is that some batch ques have a minimum number of nodes or processors that can be used. On these machine you may have to change the queue (in some way similar to the above for bluefire) and possibly the time-limits of the job, to get it to run in the batch que.

Another way to get around this problem is to run the job interactively using USE\_MPISERIAL so that you don't submit the job to the batch que. For single point mode you also may want to consider using a smaller workstation or cluster, rather than a super-computer, because you can't take advantage of the multi-processing power of the super-computer anyway.

## Running Supported Single-point/Regional Datasets

In addition to PTS\_MODE the CLM supports running using single-point or regional datasets that are customized to a particular region. In the section below we tell the user how to create their own dataset, but we also support a small number of single-point and regional datasets that are ready to setup and run in the CESM modeling system.

To get the list of supported dataset resolutions see the method given in the section on use of CLM\_PT1\_NAME, which results in the following:

```
build-namelist - valid values for res (Horizontal resolutions
Note: 0.5x0.5 and 0.33x0.33 are only used for CLM tools):
      Values: default 512x1024 128x256 64x128 48x96 32x64 8x16 94x192 0.23x0.31 \
0.47x0.63 0.9x1.25 1.9x2.5 2.5x3.33 4x5 10x15 5x5_amazon 1x1_tropicAtl 1x1_camdenNJ \
1x1_vancouverCAN 1x1_mexicocityMEX 1x1_asphaltjungleNJ 1x1_brazil 1x1_urbanc_alpha 1x1_numaIA \
1x1_smallvilleIA 0.5x0.5 0.33x0.33
      Default = 1.9x2.5
      (NOTE: resolution and mask and other settings may influence what the default is)
```

The resolution names that have an underscore in them ("\_") are all single-point or regional resolutions. To run with the supported single-point and regional datasets, you setup a simulation for the "pt1\_pt1" resolution and give the short-name for the file to use in the env\_conf.xml file.

To run for the Brazil test site do the following:

**Example 5-1. Example of running CLM over a single-point test site in Brazil with the default Qian atmosphere data forcing.**

```
> cd scripts
> ./create_newcase -case testSPDATASET -res pt1_pt1 -compset I \
-mach bluefire
> cd testSPDATASET
# Configure to run for the test site
> set SITE=1x1_brazil
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-sitespf_pt $SITE"
> ./xmlchange -file env_conf.xml -id CLM_PT1_NAME -val $SITE
```

Then configure, build and run normally.

Then to run for the urban Mexico City Mexico test site that also has atmosphere forcing data, but to run it with the Qian forcing data, but over the period for which it's own forcing data is provided do the following:

**Example 5-2. Example of running CLM over the single-point of Mexicocity Mexico with the default Qian atmosphere data forcing.**

```
> cd scripts
> ./create_newcase -case testSPDATASET -res pt1_pt1 -compset I \
-mach bluefire
> cd testSPDATASET
# Set a variable to the site you want to use (as it's used several times below)
> set SITE=1x1_mexicocityMEX
# Configure to run for the urban test site
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-sitespf_pt $SITE"
> ./xmlchange -file env_conf.xml -id CLM_PT1_NAME -val $SITE
# Set DATM prescribed aerosols to single-point dataset
# Will then use the dataset with just the point for this $SITE
> ./xmlchange -file env_conf.xml -id DATM_PRESAERO -val pt1_pt1
#
# Set some of the settings that are particular to this site, by values contained
# in the XML database. For some sites, or for new sites this information won't be
# stored. And the queryDefaultNamelist.pl command will abort.
#
# Set DATM start and end range (optional just to run over the same years that
# atmospheric forcing data is available for this site)
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_START -val \
'../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist default_settings -silent -var datm_cycle_beg_year -justvalue'
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_END -val \
'../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist default_settings -silent -var datm_cycle_end_year -justvalue'
```

Then configure, build and run normally.

**Important:** Just like PTS\_MODE above, By default it sets up to run with USE\_MPISERIAL (in the env\_build.xml file) turned on, which allows you to run the model interactively. On some machines this mode is NOT supported and you may need to change it to FALSE before you are able to build.

## Warning

See the Section called *Warning about Running with a Single-Processor on a Batch Machine* for a warning about running single-point jobs on batch machines.

**Note:** Note, that when running a pt1\_pt1 resolution the number of processors is automatically set to one. When running a single grid point you can only use a single processor. You might also want to set the "env\_conf" variable: USE\_MPISERIAL to TRUE so that you can also run interactively without having to use mpi to start up your job.

## Running Supported Single-point Datasets that have their own Atmospheric Forcing

Of the supported single-point datasets we have three that also have atmospheric forcing data that go with them: Mexico City (Mexico), Vancouver, (Canada, British Columbia), and urbanc\_alpha (test data for an Urban inter-comparison project). Mexico city and Vancouver also have "#ifdef" in the source code for

them to work with modified urban data parameters that are particular to these locations. They can be turned on by using the CLM\_CONFIG\_OPTS env\_conf.xml variable to set the "-sitespf\_pt" option in the **CLM configure**. To turn on the atmospheric forcing for these datasets, you set the env\_conf.xml DATM\_MODE variable to "CLM1PT", and then the atmospheric forcing datasets will be used for the point picked.

When running with datasets that have their own atmospheric forcing you need to be careful to run over the period that data is available. If you have at least one year of forcing it will cycle over the available data over and over again no matter how long of a simulation you run. However, if you have less than a years worth of data (or if the start date doesn't start at the beginning of the year, or the end date doesn't end at the end of the year) then you won't be able to run over anything but the data extent. In this case you will need to carefully set the RUN\_STARTDATE, START\_TOD and STOP\_N/STOP\_OPTION variables for your case to run over the entire time extent of your data. For the supported data points, these values are in the XML database and you can use the **queryDefaultNamelist.pl** script to query the values and set them for your case (they are set for the three urban test cases: Mexicocity, Vancouver, and urbanc\_alpha).

In the example below we will show how to do this for the Vancouver, Canada point.

### **Example 5-3. Example of running CLM over the single-point of Vancouver Canada with supplied atmospheric forcing data for Vancouver.**

```
> cd scripts
# Create a case at the single-point resolutions
> ./create_newcase -case testSPDATASETnAtmForcing -res pt1_pt1 -compset I \
-mach bluefire
> cd testSPDATASETnAtmForcing
# Set a variable to the site you want to use (as it's used several times below)
> set SITE=lx1_vancouverCAN
> ./xmlchange -file env_conf.xml -id CLM_CONFIG_OPTS -val "-sitespf_pt $SITE"
# Now set the CLM single-point variable to the site name
> ./xmlchange -file env_conf.xml -id CLM_PT1_NAME -val $SITE
# Set the aerosols to use the single-point dataset for 2000 conditions
# You could also use the default global dataset, but running would be a bit slower
> ./xmlchange -file env_conf.xml -id DATM_MODE -val CLM1PT
# Set the coupling frequency to once an hour
> ./xmlchange -file env_conf.xml -id ATM_NCPL -val 24
# Set the standard namelist options for an urban test site
> ./xmlchange -file env_conf.xml -id CLM_NML_USE_CASE -val stdurbpt
# Set many of the settings that are particular to this site, by values contained
# in the XML database. For some sites, or for new sites this information won't be
# stored. And the queryDefaultNamelist.pl command will abort.
#
# Set the start date
> setenv RUN_STARTDATE \
'../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist_default_settings -silent -var run_startdate -justvalue'
> setenv STARTDATE `echo $RUN_STARTDATE | sed s/-//g'
> @ START_YEAR = $STARTDATE / 10000
> ./xmlchange -file env_conf.xml -id RUN_STARTDATE -val $RUN_STARTDATE
# Set the run length and start time of day
> ./xmlchange -file env_run.xml -id STOP_OPTION \
-val '../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist_seq_timemgr_inparm -silent -var stop_option -justvalue'
> setenv STOP_N \
'../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist_seq_timemgr_inparm -silent -var stop_n -justvalue'
> ./xmlchange -file env_run.xml -id STOP_N -val $STOP_N
> ./xmlchange -file env_run.xml -id START_TOD \
-val '../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist_seq_timemgr_inparm -silent -var start_tod -justvalue'
# Set DATM start and end range...
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_START -val
'../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist_default_settings -silent -var datm_cycle_beg_year -justvalue'
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_END -val
```

```
'../../../../models/lnd/clm/bld/queryDefaultNamelist.pl -res $SITE \
-namelist default_settings -silent -var datm_cycle_end_year -justvalue'
# Set the User namelist to set the output frequencies of the history files
# Setting the stdurbpt use-case option create three history file streams
# The frequencies and number of time-samples needs to be set
> cat << EOF > user_nl_clm
&clm_inparm
hist_mfilt = $STOP_N,$STOP_N,$STOP_N
hist_nhtfrq = -1,-1,-1
/
EOF
# Set align year to start year as defined above
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_ALIGN -val $START_YEAR
# Set DATM prescribed aerosols to single-point dataset
# Will then use the dataset with just the point for this site
> ./xmlchange -file env_conf.xml -id DATM_PRESAERO -val pt1_pt1
> ./configure -case
```

## Caution

If you don't set the start-year and run-length carefully as shown above the model will abort with a "dtlimit error" in the atmosphere model (see bug 1110 in the models/lnd/clm/doc/KnownLimitations (./KnownLimitations) file for documentation on this). Since, the forcing data for this site (and the MexicoCity site) is less than a year, the model won't be able to run for a full year. The 1x1\_urbanc\_alpha site has data for more than a full year, but neither year is complete hence, it has the same problem (see the problem for this site above).

**Important:** Just like PTS\_MODE above, By default it sets up to run with USE\_MPISERIAL (in the env\_build.xml file) turned on, which allows you to run the model interactively. On some machines this mode is NOT supported and you may need to change it to FALSE before you are able to build.

## Warning

See the Section called *Warning about Running with a Single-Processor on a Batch Machine* for a warning about running single-point jobs on batch machines.

**Note:** Note, that when running a pt1\_pt1 resolution the number of processors is automatically set to one. When running a single grid point you can only use a single processor. You might also want to set the "env\_conf" variable: USE\_MPISERIAL to TRUE so that you can also run interactively without having to use mpi to start up your job.

# Creating your own single-point/regional surface datasets

The file: Quickstart.userdatasets (./Quickstart.userdatasets) in the models/lnd/clm/doc directory gives guidelines on how to create and run with your own single-point or regional datasets. Below we reprint the above guide.

```

Quick-Start to using your own datasets in clm4
=====
Assumptions: You are already familiar with the use of the cp17 scripts
for creating cases to run with "standalone" clm. See the
Quickstart.GUIDE and the README files and documentation in
the scripts directory for more information on this process.
We also assume that the env variable $CSMDATA points to the
location of the standard datasets for your machine
(/fis/cgd/cseg/csm/inputdata on bluefire). We also assume that the
following variables are used to point to the appropriate
values that you want to use for your case. Mask is included
as part of your resolution for your case, and SIM_YEAR and
SIM_YEAR_RANGE will be set appropriately for the particular use
case that you choose for your compset (i.e. 1850_control,
20thC_transient etc.).

SIM_YEAR ----- Simulation year      (i.e. 1850, or 2000)
SIM_YEAR_RANGE -- Simulation year range (i.e. constant, or 1850-2000)
MASK ----- Land mask          (i.e. navy, USGS, or gx1v6)

Process:

0.) Why do this?

An alternative to the steps below, is to create your case, and hand-edit the
relevant namelists as appropriate with your own datasets. One reason for
the process below is so that we can do automated testing on dataset inclusion.
But, it also provides the following functionality to the user:
  a.) New cases with the same datasets only require a small change to
      env_conf.xml and env_run.xml (steps 5,6, and 8)
  b.) You can clone new cases based on a working case, without having to
      hand-edit all of the namelists for the new case in the same way.
  c.) The process will check for the existence of files when cases are
      configured so you can have the scripts check that datasets exist
      rather than finding out at run-time after submitted to batch.
  d.) The process checks for valid namelists, and makes it less likely
      for you to put an error or typo in the namelists.
  e.) The *.input_data_list files will be accurate for your case,
      you can use the check_input_data script to do queries on the files.
  f.) Your dataset names will be closer to standard clm (with the exception of creation dates).
  g.) The regional extraction script (see 3.b below) will automatically create
      files with names following this convention.

1.) Create your own dataset area -- link it to standard dataset location

Create a directory to put your own datasets (such as /ptmp/$USER/my_inputdata).
Use the script link_dirtree to link the standard datasets into this location.
If you already have complete control over the datasets in $CSMDATA -- you
can skip this step.

setenv MYCSMDATA /ptmp/$USER/my_inputdata
scripts/link_dirtree $CSMDATA $MYCSMDATA

If you do this you can find the files you've added with...

find $MYCSMDATA -type f -print

and you can find the files that are linked to the standard location with...

find $MYCSMDATA -type l -print

2.) Establish a "user dataset identifier name" string

You need a unique identifier for your datasets for a given resolution,
mask, area, simulation-year, and simulation year-range. The identifier
can be any string you want -- but we have the following suggestions:

Suggestions for global grids:
```

```

setenv MYDATAID ${degLat}x${degLon}

Suggestions for regional grids: either give the number of points in the grid

setenv MYDATAID nxmpt_citySTATE
setenv MYDATAID nxmpt_cityCOUNTRY
setenv MYDATAID nxmpt_regionCOUNTRY
setenv MYDATAID nxmpt_region

or give the total size of the gridcells

setenv MYDATAID nxmdeg_citySTATE
setenv MYDATAID nxmdeg_cityCOUNTRY

for example: setenv MYDATAID 10x15 -- global 10x15 grid
            setenv MYDATAID 1x1pt_boulderCO -- single-point for Boulder CO
            setenv MYDATAID 5x5pt_boulderCO -- 5x5 region around Boulder CO
            setenv MYDATAID 1x1deg_boulderCO - 1x1 degree region around Boulder CO
            setenv MYDATAID 13x12pt_f19_alaskaUSA1 - 13x12 gridcells from f19
            (1.9x2.5) global resolution over Alaska

```

3.) Add your own datasets in the standard locations in that area

3.a) Create datasets using the standard tools valid for any specific points

Use the tools in models/lnd/clm/tools to create new datasets. Tools such as: mkgriddata, mksurfdata, mkdatadomain, and the regridding tools in ncl\_scripts

(see the models/lnd/clm/bld/namelist\_files/namelist\_defaults\_usr\_files.xml for the exact syntax for all files).

```

surfdata: copy files into:
$MYCSMDATA/lnd/clm2/surfdata/surfdata_${MYDATAID}_simyr${SIM_YEAR}.nc
fatmgrid: copy files into:
$MYCSMDATA/lnd/clm2/griddata/griddata_${MYDATAID}.nc
fatmlndfrc: copy files into:
$MYCSMDATA/lnd/clm2/griddata/fracdata_${MYDATAID}_${MASK}.nc
domainfile: copy files into:
$MYCSMDATA/atm/datm7/domain.clm/domain.lnd.${MYDATAID}_${MASK}.nc

```

3.b) Use the regional extraction script to get regional datasets from the global ones  
 Use the getregional\_datasets.pl script to extract out regional datasets of interest.  
 Note, the script works on all files other than the "finidat" file as it's a 1D vector file.

For example, Run the extraction for data from 52-73 North latitude, 190-220 longitude  
 that creates 13x12 gridcell region from the f19 (1.9x2.5) global resolution over Alaska.

```

cd models/lnd/clm/tools/ncl_scripts
./getregional_datasets.pl -sw 52,190 -ne 73,220 -id $MYDATAID \
-myccsmda $MYCSMDATA

```

Repeat this process if you need files for multiple sim\_year, and sim\_year\_range values.

4.) Setup your case

Follow the standard steps for executing "scripts/create\_newcase" and customize your case as appropriate.

i.e.

```

./create_newcase -case my_userdataset_test -res pt1_pt1 -compset I1850 \
-mach bluefire

```

The above example implies that: MASK=gx1v6, SIM\_YEAR=1850, and SIM\_YEAR\_RANGE=constant.  
 5.) Edit the env\_run.xml in the case to point to your new dataset area

Edit DIN\_LOC\_ROOT\_CSMDATA in env\_run.xml to point to \$MYCSMDATA

```

./xmlchange -file env_run.xml -id DIN_LOC_ROOT_CSMDATA -val $MYCSMDATA

```

6.) Edit the env\_conf.xml in the case to point to your user dataset identifier name.

Edit CLM\_USRDATA\_NAME to point to \$MYDATAID

```

./xmlchange -file env_conf.xml -id CLM_USRDATA_NAME -val $MYDATAID

```

```

./xmlchange -file env_conf.xml -id CLM_PT1_NAME      -val $MYDATAID
7.) Configure the case as normal
./configure -case
8.) Run your case as normal

```

## Using getregional\_datasets.pl to get a complete suite of single-point/regional surface datasets from global ones

Use the regional extraction script to get regional datasets from the global ones. The getregional\_datasets.pl script to extract out regional datasets of interest. Note, the script works on all files other than the "finidat" file as it's a 1D vector file. The script will extract out a block of gridpoints from all the input global datasets, and create the full suite of input datasets to run over that block. The input datasets will be named according to the input "id" you give them and the id can then be used as input to CLM\_USRDATA\_NAME to create a case that uses it. See the section on CLM Script Configuration Items for more information on setting CLM\_USRDATA\_NAME (in Chapter 1). The list of files extracted by their name used in the namelists are: fatmgrd, fatmlndfrc, fsurdat, fpftdyn, flndtopo, stream\_fldfilename\_ndep, and the DATM files domainfile, and faerdep. For more information on these files see the Table on required files.

The alternatives to using this script are to use PTS\_MODE, discussed earlier, to use PTCLM discussed in the next chapter, or creating the files individually using the different file creation tools (given in the Tools Chapter). Creating all the files individually takes quite a bit of effort and time. PTS\_MODE has some limitations as discussed earlier, but also as it uses global files, is a bit slower when running simulations than using files that just have the set of points you want to run over. Another advantage is that once you've created the files using this script you can customize them if you have data on this specific location that you can replace with what's already in these files.

The script requires the use of both "Perl" and "NCL". See the NCL Script section in the Tools Chapter on getting and using NCL and NCL scripts. The main script to use is a Perl script which will then in turn call the NCL script that actually creates the output files. The ncl script gets its settings from environment variables set by the perl script. To get help with the script use "-help" as follows:

```

> cd models/lnd/clm/tools/ncl_scripts
> ./getregional_datasets.pl -help

```

The output of the above is:

```

SYNOPSIS
    getregional_datasets.pl [options]      Extracts out files for a single box region from the \
global                                         grid for the region of interest. Choose a box determined by
                                              the NorthEast and SouthWest corners.

OPTIONS
    -debug [or -d]                         Just debug by printing out what the script would do.
                                              This can be useful to find the size of the output area.
    -help [or -h]                           Print usage to STDOUT.
    -mask "landmask"                        Type of land-mask (i.e. navy, gx3v7, gx1v6 etc.) (default gx1v6)
    \
    -mycsmdata "dir"                       Root directory of where to put your csmdata.
                                              (default /home/erik/inputdata or value of CSMDATA env variable)

```

```

-mydataid "name" [or -id]      Your name for the region that will be extracted.      \
(REQUIRED)
                                         Recommended name: grid-size_global-resolution_location  \
(?x?pt_f??_????)          (i.e. 12x13pt_f19_alaskaUSA for 12x13 grid cells from the f19  \
global resolution over Alaska)                                                 \
-NE_corner "lat,lon" [or -ne] North East corner latitude and longitude           \
(REQUIRED)
-nomv                         Do NOT move datasets to final location, just leave them in  \
current directory
-res "resolution"             Global horizontal resolution to extract data from (default  \
1.9x2.5).
-rcp "pathway"                Representative concentration pathway for future scenarios
Only used when simulation year range ends in a future
year, such as 2100.
(default -999.9).
-sim_year "year"              Year to simulate for input datasets (i.e. 1850, 2000) (default  \
2000)
(default 2000)
-sim_yr_rng "year-range"      Range of years for transient simulations
(i.e. 1850-2000, 1850-2100, or constant) (default constant)
-SW_corner "lat,lon" [or -sw] South West corner latitude and longitude           \
(REQUIRED)
-verbose [or -v]               Make output more verbose.

```

The *required* options are: `-id`, `-ne`, and `-se`, for the output identifier name to use in the filenames, latitude and longitude of the Northeast corner, and latitude and longitude of the SouthEast corner (in degrees). Options that specify which files will be used are: `-mask`, `-res`, `-rcp`, `-sim_year`, and `-sim_yr_rng` for the land-mask to use, global resolution name, representative concentration pathway for future scenarios, simulation year, and simulation year range. The location of the input and output files will be determined by the option `-mycsmdata` (can also be set by using the environment variable `$CSMDATA`). If you are running on a machine like at NCAR where you do NOT have write permission to the CESM inputdata files, you should use the `scripts/link_dirtree` script to create soft-links of the original files to a location that you can write to. This way you can use both your new files you created as well as the original files and use them from the same location.

The remaining options to the script are `-debug`, and `-verbose`. `-debug` is used to show what would happen if the script was run, without creating the actual files. `-verbose` adds extra log output while creating the files so you can more easily see what the script is doing.

For example, Run the extraction for data from 52-73 North latitude, 190-220 longitude that creates 13x12 gridcell region from the f19 (1.9x2.5) global resolution over Alaska.

#### **Example 5-4. Example of running getregional\_datasets.pl to get datasets for a specific region over Alaska**

```

> cd scripts
# First make sure you have a inputdata location that you can write to
# You only need to do this step once, so you won't need to do this in the future
> setenv MYCSMDATA $HOME/inputdata      # Set env var for the directory for input data
> ./link_dirtree $CSMDATA $MYCSMDATA
> cd ../models/lnd/clm/tools/ncl_scripts
> ./getregional_datasets.pl -sv 52,190 -ne 73,220 -id 13x12pt_f19_alaskaUSA -mycsmdata $MYCSMDATA

```

Repeat this process if you need files for multiple sim\_year, resolutions, land-masks, and sim\_year\_range values.

## Warning

See the Section called *Warning about Running with a Single-Processor on a Batch Machine* for a warning about running single-point jobs on batch machines.

**Note:** See the Section called *Managing Your Own Data-files* in Chapter 3 for notes about managing your data when using `link_dirtree`.

Now to run a simulation with the datasets created above, you create a single-point case, and set CLM\_USRDAT\_NAME to the identifier used above. Note that in the example below we set the number of processors to use to one (-pecount 1). For a single point, you should only use a single processor, but for a regional grid, such as the example below you could use up to the number of grid points (12x13=156 processors).

**Example 5-5. Example of using CLM\_USRDAT\_NAME to run a simulation using user datasets for a specific region over Alaska**

```
> cd scripts
# Create the case and set it to only use one processor
> ./create_newcase -case my_userdataset_test -res pt1_pt1 -compset I1850 \
-mach bluefire
> cd my_userdataset_test/
> ./xmlchange -file env_run.xml -id DIN_LOC_ROOT_CSMDATA -val $MYCSMDATA
> ./xmlchange -file env_conf.xml -id CLM_USRDAT_NAME -val 13x12pt_f19_alaskaUSA
> ./xmlchange -file env_conf.xml -id CLM_BLDNML_OPTS -val '-mask gx1v6'
> ./xmlchange -file env_conf.xml -id CLM_PT1_NAME -val 13x12pt_f19_alaskaUSA
> ./configure -case
```

## Running with your own atmosphere forcing

Here we want to run with our own customized datasets for CLM as well as running with our own supplied atmosphere forcing datasets. Thus we effectively combine the information from the Section called *Running Supported Single-point Datasets that have their own Atmospheric Forcing* with the Section called *Creating your own single-point/regional surface datasets*. First we need to follow the procedures in the Section called *Running Supported Single-point Datasets that have their own Atmospheric Forcing* to come up with CLM datasets that are customized for our point or region in question. This includes running `link_dirtree` to create a directory location where you can add your own files to it. Next, set DATM\_MODE to "CLM1PT" and CLM\_PT1\_NAME and CLM\_USRDAT\_NAME to the id of the data you created. To see a list of what the filenames need to be see the section on setting CLM\_USRDAT\_NAME.

Next we need to setup the atmosphere forcing data in NetCDF format that can be read by DATM. There is a list of eight variables that are expected to be on the input files with the names and units on the following table (in the table TDEW and SHUM are optional fields that can be used in place of RH). In the table we also list which of the fields are required and if not required what the code will do to replace them. If the names of the fields are different or the list is changed from the standard list of eight fields:

FLDS, FSDS, PRECTmms, PSRF, RH, TBOT, WIND, and ZBOT, the resulting streams file will need to be modified to take this into account (see an example streams file for this in Example 5-7 below).

**Table 5-1. Atmosphere Forcing Fields**

<b>Short-name</b>	<b>Description</b>	<b>Units</b>	<b>Required?</b>	<b>If NOT required how replaced</b>
FLDS	incident longwave (FLDS)	W/m2	No	calculates based Temperature, Pressure and Humidity
FSDS	incident solar (FSDS)	W/m2	Yes	-
FSDSdif	incident solar (FSDS) diffuse	W/m2	No	based on FSDS
FSDSdir	incident solar (FSDS) direct	W/m2	No	based on FSDS
PRECTmms	precipitation (PRECTmms)	mm/s	Yes	-
PSRF	pressure at the lowest atm level (PSRF)	Pa	No	assumes standard-pressure
RH	relative humidity at the lowest atm level (RH)	%	No	can be replaced with SHUM or TDEW
SHUM	specific humidity at the lowest atm level	kg/kg	Optional in place of RH	can be replaced with RH or TDEW
TBOT	temperature at the lowest atm level (TBOT)	K (or can be C)	Yes	-
TDEW	dew point temperature	K (or can be C)	Optional in place of RH	can be replaced with RH or SHU
WIND	wind at the lowest atm level (WIND)	m/s	Yes	-
ZBOT	observational height	m	No	assumes 30 meter

All of the variables should be dimensioned: time, lat, lon, with time being the unlimited dimension. The coordinate variable "time" is also required with CF-compliant units in days, hours, minutes, or seconds. It can also have a calendar attribute that can be "noLeap" or "gregorian". Normally the files will be placed in the: \$MYCSMDATA/atm/datm7/CLM1PT\_data/\$MYUSRDAT directory with separate files per month called YYYY-MM.nc where YYYY-MM corresponds to the four digit year and two digit month with a dash in-between. You also need a domain file that gives the coordinate information for the data that should be placed in: \$MYCSMDATA/atm/datm7/domain.lnd.\$MYUSRDAT\_USGS.nc.

### Example 5-6. Example of setting up a case with your own atmosphere forcing

```
> cd scripts
# First make sure you have a inputdata location that you can write to
# You only need to do this step once, so you won't need to do this in the future
> setenv MYCSMDATA $HOME/inputdata      # Set env var for the directory for input data
> ./link_dirtree $CSMDATA $MYCSMDATA
# Next create and move all your datasets into $MYCSMDATA with id $MYUSRDAT
# See above for naming conventions

# Now create a single-point case
> ./create_newcase -case my_atmforc_test -res pt1_pt1 -compset I1850 \
-mach bluefire
> cd my_atmforc_test
# Set the data root to your inputdata directory, and set CLM_PT1_NAME and CLM_USRDATA_NAME
# to the user id you created for your datasets above
> ./xmlchange -file env_run.xml -id DIN_LOC_ROOT_CSMDATA -val $MYCSMDATA
> ./xmlchange -file env_conf.xml -id CLM_PT1_NAME -val $MYUSRDAT
> ./xmlchange -file env_conf.xml -id CLM_USRDATA_NAME -val $MYUSRDAT
# Set the land-mask to USGS, so both clm and DATM can find files
> ./xmlchange -file env_conf.xml -id CLM_BLDNML_OPTS -val '-mask USGS'
# Then set DATM_MODE to single-point mode so DATM will use your forcing datasets
# Put your forcing datasets into $MYCSMDATA/atm/datm7/CLM1PT_data/$MYUSRDAT
> ./xmlchange -file env_conf.xml -id DATM_MODE -val CLM1PT
> ./configure -case
# If the list of fields, or filenames, filepaths, or fieldnames are different
# you'll need to edit the DATM namelist streams file to make it consistent
> $EDITOR Buildconf/datm.buildnml.csh
```

## Warning

See the Section called *Warning about Running with a Single-Processor on a Batch Machine* for a warning about running single-point jobs on batch machines.

**Note:** See the Section called *Managing Your Own Data-files* in Chapter 3 for notes about managing your data when using **link\_dirtree**.

Now, we'll show an example of what the DATM streams file might look like for a case with your own forcing data with 3-hourly forcing. In this example, we'll leave off the fields: ZBOT, and FLDS so they'll be calculated as given in the Table 5-1 table above. We'll also include: FSDDsf and FSDDsdir which aren't required, and we'll use TDEW in place of RH. In this example the datafiles are in NetCDF format and contain the fields: TA, Tdew, WS, PREC, Rg, Rgdir, Rgdif, and PRESS which are translated into the DATM internal names in this streams file. There is also a domain file that has the position information for this location. The normal assumption for CLM1PT mode in the DATM is that data is hourly or half-hourly and as such is often enough that using the data on the nearest time-stamp is reasonable and as such the data is in a single streams file (see the Section called *CLM1PT mode and it's DATM settings* in Chapter 1 for more information on the default settings for DATM and how to change them. If the data is less often three to six hours -- see Example 5-7 below, where you will need to modify the time-interpolation method as well as the time stamp offsets. In the example below we also have to divide the single stream file into three files to manage the time-stamps and time interpolation algorithm for the different types of data differently.

**Example 5-7. Example of DATM streams files with your own forcing for 3-hourly data**

Precipitation streams file (`clm1PT.1x1pt_lapazMEX.precip.stream.txt` file).

```
<streamstemplate>
<stream>
    <dataSource>
        CLMNCEP
    </dataSource>
    <domainInfo>
        <variableNames>
            time      time
            xc       lon
            yc       lat
            area     area
            mask     mask
        </variableNames>
        <filePath>
            $DIN_LOC_ROOT/atm/datm7/domain.clm
        </filePath>
        <fileNames>
            domain.lnd.1x1pt_lapazMEX_navy.nc
        </fileNames>
    </domainInfo>
    <fieldInfo>
        <variableNames>
            PRECTmms PREC
        </variableNames>
        <offset>
            -5400
        </offset>
        <filePath>
            $DIN_LOC_ROOT/atm/datm7/CLM1PT_data/1x1pt_lapazMEX
        </filePath>
        <fileNames>
            2004-01.nc
            2004-02.nc
            2004-03.nc
        </fileNames>
    </fieldInfo>
    .
    .
    .
    <variableNames>
        2009-12.nc
    </fileNames>
    </fieldInfo>
</stream>
</streamstemplate>
```

Solar streams file (`clm1PT.1x1pt_lapazMEX.solar.stream.txt` file).

```
<streamstemplate>
<stream>
    <dataSource>
        CLMNCEP
    </dataSource>
    <domainInfo>
        <variableNames>
            time      time
            xc       lon
            yc       lat
            area     area
            mask     mask
        </variableNames>
        <filePath>
            $DIN_LOC_ROOT/atm/datm7/domain.clm
        </filePath>
        <fileNames>
            domain.lnd.1x1pt_lapazMEX_navy.nc
        </fileNames>
    </domainInfo>
    <fieldInfo>
        <variableNames>
            FSDS      Rg
            FSDSdir   Rgdir
            FSDSdif   Rgdif
        </variableNames>
    </fieldInfo>
</stream>
</streamstemplate>
```

```

<offset>
-10800
</offset>
<filePath>
$DIN_LOC_ROOT/atm/datm7/CLM1PT_data/1x1pt_lapazMEX
</filePath>
<fileNames>
2004-01.nc
2004-02.nc
2004-03.nc
2005-01.nc
2005-02.nc
2005-03.nc
2005-04.nc
2005-05.nc
2005-06.nc
2005-07.nc
2005-08.nc
2005-09.nc
2005-10.nc
2005-11.nc
2005-12.nc
2006-01.nc
2006-02.nc
2006-03.nc
2006-04.nc
2006-05.nc
2006-06.nc
2006-07.nc
2006-08.nc
2006-09.nc
2006-10.nc
2006-11.nc
2006-12.nc
2007-01.nc
2007-02.nc
2007-03.nc
2007-04.nc
2007-05.nc
2007-06.nc
2007-07.nc
2007-08.nc
2007-09.nc
2007-10.nc
2007-11.nc
2007-12.nc
2008-01.nc
2008-02.nc
2008-03.nc
2008-04.nc
2008-05.nc
2008-06.nc
2008-07.nc
2008-08.nc
2008-09.nc
2008-10.nc
2008-11.nc
2008-12.nc
2009-01.nc
2009-02.nc
2009-03.nc
2009-04.nc
2009-05.nc
2009-06.nc
2009-07.nc
2009-08.nc
2009-09.nc
2009-10.nc
2009-11.nc
2009-12.nc
</fileNames>
</fieldInfo>
</stream>
</streamstemplate>

```

Other fields streams file. (clm1PT.1x1pt\_lapazMEX.other.stream.txt file).

```

<streamstemplate>
<stream>
<dataSource>
CLMNCEP
</dataSource>
<domainInfo>
<variableNames>
time time
xc lon
yc lat
area area
mask mask
</variableNames>
<filePath>
$DIN_LOC_ROOT/atm/datm7/domain.clm
</filePath>
<fileNames>
domain.lnd.1x1pt_lapazMEX_navy.nc
</fileNames>
</domainInfo>
<fieldInfo>
<variableNames>
TBOT TA
TDEW Tdew
WIND WS
PSRF PRESS
</variableNames>
<offset>
-5400
</offset>
<filePath>
$DIN_LOC_ROOT/atm/datm7/CLM1PT_data/1x1pt_lapazMEX
</filePath>
<fileNames>
2004-01.nc
2004-02.nc
2004-03.nc
2005-01.nc
2005-02.nc
2005-03.nc
2005-04.nc
2005-05.nc
2005-06.nc
2005-07.nc
2005-08.nc
2005-09.nc
2005-10.nc
2005-11.nc
2005-12.nc
2006-01.nc
2006-02.nc
2006-03.nc
2006-04.nc
2006-05.nc
2006-06.nc
2006-07.nc
2006-08.nc
2006-09.nc
2006-10.nc
2006-11.nc
2006-12.nc
2007-01.nc
2007-02.nc
2007-03.nc
2007-04.nc
2007-05.nc
2007-06.nc
2007-07.nc
2007-08.nc
2007-09.nc
2007-10.nc
2007-11.nc
2007-12.nc
2008-01.nc
2008-02.nc
2008-03.nc
2008-04.nc
2008-05.nc
2008-06.nc
2008-07.nc
2008-08.nc
2008-09.nc
2008-10.nc
2008-11.nc
2008-12.nc
2009-01.nc
2009-02.nc
2009-03.nc
2009-04.nc
2009-05.nc
2009-06.nc
2009-07.nc
2009-08.nc
2009-09.nc
2009-10.nc
2009-11.nc
2009-12.nc
</fileNames>
</fieldInfo>
</stream>
</streamstemplate>

```

Example streams namelist for the above streams files:

```

&shr_strdata_nml
dataMode      = 'CLMNCEP'
domainFile   = '$DOMAINFILE'
streams      = 'clm1PT.1x1pt_lapazMEX.solar.stream.txt 1 2004 2009 ',
               'clm1PT.1x1pt_lapazMEX.precip.stream.txt 1 2004 2009 ',
               'clm1PT.1x1pt_lapazMEX.other.stream.txt 1 2004 2009 ',

```

```
'presaero.stream.txt 1 2000 2000'  
vectors      = 'null','null','null','null'  
mapmask      = 'nomask','nomask','nomask','nomask'  
mapalgo      = 'nn','nn','nn','nn'  
tintalgo     = 'coszen','nearest','linear','linear'  
taxmode      = 'cycle','cycle','cycle','cycle'  
/
```

**Note:** The example above shows the resolved namelist and streams file after **configure** has been run. In order to save this configuration for future use, you would need to edit the DATM template adding new DATM\_MODE see the Section called *Adding a new DATM\_MODE to the DATM template* in Appendix A for more information on how to do this.

We've outlined and given a few examples of using your own atmosphere forcing. In the next chapter we go into the details of using PTCLM1.110726.

# Chapter 6. How to run PTCLM1.110726

PTCLM (pronounced either as point clime or Pee-Tee clime) is a Python script to help you set up Point CLM simulations. It runs the CLM tools for you to get datasets set up, and copies them to a location you can use them according to the CLM\_USRDAT\_NAME naming convention. Then it runs `create_newcase` for you and modifies the env settings and namelist appropriately. PTCLM has a simple ASCII text file for storing basic information for your sites. We also have complete lists for AmeriFlux and Fluxnet-Canada sites, although we only have the meteorology data for one site. For other sites you will need to obtain the meteorology data and translate it to a format that the CESM DATM model can use. But, even without meteorology data PTCLM is useful to setup datasets to run with standard CLM\_QIAN data.

The original authors of PTCLM are: Daniel M. Ricciuto, Dali Wang, Peter E. Thornton, Wilfred M. Post all at Environmental Sciences Division, Oak Ridge National Laboratory (ORNL) and R. Quinn Thomas at Cornell University. It was then modified fairly extensively by Erik Kluzek at NCAR. We want to thank all of these individuals for this contribution to the CESM effort. We also want to thank the folks at University of Michigan Biological Stations (US-UMB) who allowed us to use their Fluxnet station data and import it into our inputdata repository, especially Gil Bohrer the PI on record for this site.

## Introduction to PTCLM

To get help on PTCLM1.110726 use the "--help" option as follows.

```
> cd scripts/ccsm_utils/Tools/lnd/clm/PTCLM  
> ./PTCLM.py --help
```

The output to the above command is as follows:

Here we give a simple example of using PTCLM1 for a straightforward case of running at the US-UMB Fluxnet site on bluefire where we already have the meteorology data on the machine. Note, see the Section called *Converting AmeriFlux Data for use by PTCLM* for permission information to use this data.

### Example 6-1. Example of running PTCLM1 for US-UMB on bluefire

```
setenv CSMDATA /fis/cgd/cseg/csm/inputdata  
setenv MYCSMDATA $HOME/inputdata  
setenv SITE US-UMB  
setenv MYMACH bluefire  
setenv MYCASE testPTCLM  
  
# First link the standard input files to a location you have write access  
cd scripts  
.link_dirtree $CSMDATA $MYCSMDATA  
  
# Next build all of the clm tools you will need
```

```

cd ../models/lnd/clm/tools/mksurfdata
gmake
gmake clean
cd ../mkdatadomain
gmake
gmake clean
cd ../mkgriddata
gmake
gmake clean
# next run PTCLM (NOTE -- MAKE SURE python IS IN YOUR PATH)
cd ../../../../scripts/ccsm_utils/Tools/lnd/clm/PTCLM
./PTCLM.py -m $MYMACH --case=$MYCASE --site=$SITE --csmdata=$MYCSMDATA \
--aerdepgrid --ndepgrid
# NOTE: we use --aerdepgrid --ndepgrid so that you use the global
# aerosol and Nitrogen deposition files rather than site-specific ones.
cd ../../../../$MYCASE
# Finally configure, build, and run the case as normal

```

## Guide to the options of PTCLM

There are three types of options to PTCLM1: required, configure/run-time, and dataset generation options. The three required options are the three settings that MUST be specified for PTCLM to work at all. The other settings have default values that will default to something useful. The configure/run-time options control how the simulation will be setup and run. The dataset generation options control the generation of datasets needed when PTCLM is run. Most options use a double dash "--" "longname" such as "--list", but the most common options also have a short-name with a single dash (such as -m instead of --machine).

The required options to PTCLM are: inputdata directory (-d), machine (-m) and site-name (-s). Inputdata directory is the directory where you have the CESM inputdata files, you need to have write access to this directory, so if you are running on a machine that you do NOT have write access to the standard inputdata location (such as NCAR bluefire or ORNL jaguar) you need to link the standard files to a location you do have control over. We recommend using the `scripts/link_dirtree` tool to do that. "machine" is the scripts name for the machine/compiler you will be using for your case. And finally site-name is the name of the site that you want to run for. Site-name can either be a valid `CLM_PT1_NAME` supported dataset name or a Fluxnet site name from the list of sites you are running on (see the `--sitegroupname` for more information about the site lists).

After PTCLM is run a case directory where you can then configure, build and run your CESM case as normal. It also creates a `README.PTCLM` in that directory that documents the commandline options to PTCLM that were used to create it.

After "help" the "list" option is one of the most useful options for getting help on using PTCLM. This option gives you information about some of the other options to PTCLM. To get a list of the machine, sites, and compsets that can be used for PTCLM use the "--list" option as follows.

```

> cd scripts/ccsm_utils/Tools/lnd/clm/PTCLM
> ./PTCLM.py --list

```

The output to the above command is as follows:

```
----- PTCLM version \
PTCLM0.5_scripts_brnchtag_cesm1_0_4_n01_PTCLM1_110726-----
./PTCLM.py --list

OPTIONS:

Site name:      list

CESM Component set:      list
CESM machine:      none
Root CLM directory:      /cesmweb/html/models/cesm1.0/clm
Case name:      list_list
Case directory:      /cesmweb/html/models/cesm1.0/clm/scripts
Accelerated Decomposition mode:      False
Exit spinup mode:      False
Final spin up:      False
Using QIAN climate inputs      False
Finidat file:      <none>
Sites group name:      EXAMPLE
Use preexisting point data:      False
** Surface data file will be built using site-level data when available unless otherwise specified \
**
Extract PFT data from gridded files:  False
Extract soil data from gridded files:  False

Valid Machines: ['bluefire', 'brutus_po', 'brutus_pm', 'brutus_io', 'brutus_im', 'edinburgh_lahey', \
'edinburgh_pgi', 'edinburgh_intel', 'franklin', 'hadley', 'hector', 'hopper', 'hopp2', \
'intrepid', 'jaguar', 'jaguarpf', 'chester', 'kraken', 'lynx_pgi', 'lynx_pathscales', 'midnight', \
'pleiades', 'pleiades_wes', 'prototype_atlas', 'prototype_hera', 'prototype_columbia', \
'prototype_frost', 'prototype_nyblue', 'prototype_ranger', 'prototype_ubgl', 'generic_ibm', \
'generic_xt', 'generic_linux_pgi', 'generic_linux_lahey', 'generic_linux_intel', \
'generic_linux_pathscales', 'generic_darwin_intel', 'generic_darwin_pgi']

Valid Compsets: ['I_TEST_2003', 'I_2000', 'I_1850', 'I_1948-2004', 'I_1850-2000', 'I_TEST_2003_CN', \
'I_2000_CN', 'I_1850_CN', 'I_1850_SPINUP_3HrWx_CN', 'I_1948-2004_CN', 'I_1850-2000_CN', \
'I_RCP8.5_CN', 'I_RCP6.0_CN', 'I_RCP4.5_CN', 'I_RCP2.6_CN']

Open Site data file: PTCLM_sitedata/EXAMPLE_sitedata.txt

site = site_code name: name                                     Region: state
site =     US-UMB name: UMBS                                    Region: MI

Supported CLM1PT name dataset names are:

site = 1x1_tropicAtl
site = 1x1_camdenNJ
site = 1x1_vancouverCAN
site = 1x1_mexicocityMEX
site = 1x1_asphaltjungleNJ
site = 1x1_brazil
site = 1x1_urbanc_alpha
site = 1x1_numaIA
site = 1x1_smallvilleIA
```

## Overview on using PTCLM

### Steps in running PTCLM

1. Setup Inputdata directory with write access (use **link\_dirtree** script)

You need to setup an inputdata directory where you have write access to it. Normally, for NCAR machines the data is on an inputdata where the user does NOT have write access to it. A way that you can get around this is to use the **link\_dirtree** script to create softlinks from the normal location to a location you have write access to. So for example on bluefire:

```
> setenv CSMDATA /fs/cgd/csm/inputdata
> setenv MYCSMDATA $HOME/inputdata
> mkdir $MYCSMDATA
> cd scripts
> ./link_dirtree $CSMDATA $MYCSMDATA
```

See the Section called *Managing Your Own Data-files* in Chapter 3 for more information on this.

2. Build the CLM tools

Next you need to make sure all the CLM FORTRAN tools are built.

```
> cd models/lnd/clm/tools/mkgriddata
> gmake
> gmake clean
> cd ../mkdatadomain
> gmake
> gmake clean
> cd ../mksurfdata
> gmake
> gmake clean
```

3. Run PTCLM

Next you actually run PTCLM1 which does the different things listed below:

- a. PTCLM names your case based on your input

PTCLM1 names your case based on the input you give to it.

```
[Prefix_]SiteCode_Compset[_QIAN] [_spinuptype]
Where:
Prefix is from the caseidprefix option (or blank if not used).
SiteCode is the site name you entered with the -s option.
Compset is the compset name you entered with the -c option.
_QIAN is part of the name only if the useQIAN is used.
_spinuptype is part of the name if one of: ad_spinup, exit_spinup, or
final_spinup is used, and the exact spinup name chosen is used.
```

For example, the casename for the following will be:

```
> cd scripts
> ./PTCLM.py -m bluefire -s US-UMB -d $MYCSMDATA -c I_2000_CN --ad_spinup --useQIAN
```

"US-UMB\_I\_2000\_CN\_QIAN\_ad\_spinup".

- b. PTCLM creates datasets for you

It will populate \$MYCSMDATA with new datasets it creates using the CLM tools.

- c. If a transient compset and PTCLM1 finds a \_dynpftdata.txt file

If you are running a transient compset (such as the "I\_1850-2000\_CN" compset) AND you there is a file in the PTCLM\_sitedata directory under the PTCLM directory called \$SITE\_dynpftdata.txt it will use this file for the land-use changes. Otherwise it will

leave land-use constant, unless you use the pftgrid option so it uses the global dataset for landuse changes. See the Section called *Dynamic Land-Use Change Files for use by PTCLM* for more information on this. There is a sample transient dataset called US-Hal\_dynpftdata.txt. Transient compsets, are compsets that create transient land-use change and forcing conditions such as: 'I\_1850-2000', 'I\_1850-2000\_CN', 'I\_RCP8.5\_CN', 'I\_RCP6.0\_CN', 'I\_RCP4.5\_CN', or 'I\_RCP2.6\_CN'.

d. PTCLM creates a pft-physiology for you

PTCLM1 will create a local copy of the pft-physiology specific for your site that you could then customize with changes specific for that site.

e. PTCLM creates a README.PTCLM for you

PTCLM1 will create a simple text file with the command line for it in a file called README.PTCLM in the case directory it creates for you.

4. Customize, configure, build and run case as normal

You then customize your case as you would normally. See the Chapter 1 chapter for more information on doing this.

## Details on the options of PTCLM

Next we discuss the configure and run-time options, dividing them up into configure, spinup, and run-time options.

Configure options include:

```
-c MYCOMPSET, --compset=MYCOMPSET
--caseidprefix=MYCASEID
--cesm_root=BASE_CESM
--namelist=NAMELIST
--rmold
--scratchroot=SCRATCHROOT
--sitegroupname=SITEGROUP
--QIAN_tower_yrs
--useQIAN
```

### --compset

The "-c" option is the most commonly used option after the required options, as it specifies the CESM scripts component set to use with PTCLM1. The default compset is the "ICN" compset with CN on for present day conditions.

### --caseidprefix

This option gives a prefix to include in the casename when the case is created, in case you want to customize your casenames a bit. By default, casenames are figured out based on the other options. The argument to this option can either be a name to prefix casenames with and/or a pathname to include. Hence, if you want cases to appear in a specific directory you can give the pathname to that directory with this option.

**--cesm\_root**

This option is for running PTCLM1 with a different root directory to CESM than the version PTCLM exists in. Normally you do NOT need to use this option.

**--namelist**

This option adds any items given into the CLM `user_nl_clm` namelist. This allows you to add customizations to the namelist before the `clm.buildnml.csh` file is created for the case.

**--rmold**

This option will remove an old case directory of the same name if one exists. Otherwise, if an old case directory already exists and you try to run PTCLM it will return with an error.

**--scratchroot**

This option is ONLY valid when using one of the generic machines (the -m option). This passed onto `create_newcase` and gives the location where cases will be built and run.

**--sitegroupname**

In the PTCLM directory there is a subdirectory "PTCLM\_sitedata" that contains files with the site, PFT and soil data information for groups of sites. These site groups are all separate ASCII files with the same prefix followed by a "\_\*data.txt" name. See the Section called *PTCLM Group Site Lists* for more information on these files. By default we have provided three different valid group names:

EXAMPLE  
AmeriFlux  
Fluxnet-Canada

The EXAMPLE is the group used by default and ONLY includes the US-UMB site as that is the only site we have data provided for. The other two site groups include the site information for all of both the AmeriFlux and Fluxnet-Canada sites. You can use the "sitegroupname" option to use one of the other lists, or you can create your own lists using the EXAMPLE file as an example. Your list of sites could be real world locations or could be theoretical "virtual" sites given to exercise CLM on differing biomes for example. Note, see the Section called *Converting AmeriFlux Data for use by PTCLM* with permission information to use the US-UMB data.

**--useQIAN**

This option says to use the standard CLM global Qian T62 atmospheric forcing rather than any tower site forcing data available. Otherwise, PTCLM will try to find tower forcing data for the specific site entered.

**--QIAN\_tower\_yrs**

This option is used with the "useQIAN" option to set the years to cycle over for the Qian data. In this case Qian atmospheric forcing will be used, but the simulation will run over the same years that tower site is available for this site.

Spinup options include:

- coldstart
- ad\_spinup
- exit\_spinup
- final\_spinup
- finidat=FINIDAT

The spinup options enable the different CN spinup modes, but also set the run length. The coldstart option says to startup with OUT an initial condition file, while the finidat option explicitly gives the initial condition file to use. Obviously, the different spinup options can NOT be used together, nor can the coldstart and finidat options be either.

#### --coldstart

This option ensures that a cold-start will be done with arbitrary initial conditions.

#### --ad\_spinup

This option enables the accelerated decomposition mode when a CN compset is used. It also sets the run-length as given in the example for running exit spinup in the Section called *Spinning up the Satellite Phenology Model (CLMSP spinup)* in Chapter 4. SEE WARNING BELOW.

#### --exit\_spinup

This option enables the exit spinup mode when a CN compset is used. It also sets the run-length to a year just as given in the example for running exit spinup in the Section called *Spinning up the Satellite Phenology Model (CLMSP spinup)* in Chapter 4. SEE WARNING BELOW.

#### --final\_spinup

This option sets the run length as given in the example for a final spinup in Example 4-5. This option can be used for any compset.

#### --finidat

This option sets the initial condition file to startup the simulation from.

Run-time options include:

```
--debug
--run_n=MYRUN_N
--run_units=MYRUN_UNITS
--stdurbpt
```

#### --debug

This option tells PTCLM to echo what it would do if it were run, but NOT actually run anything. So it will show you the dataset creation commands it would use. It does however, run **create\_newcase**, but then it only displays the **xmlchange** commands and changes that it would do. Also note that if you give the "--rmold" option it won't delete the case directory beforehand. Primarily this is intended for debugging the operation of PTCLM.

#### --run\_n

This option along with run\_units is used to set the length for the simulation. "run\_n" is the number of units to use. The default run length depends on the site, compset, and configuration and for example if a "spinup" option is selected.

**--run\_units**

This option is the units of time to use for the length of the simulation. It is used along with "run\_n" to set the length of the simulation. The default run length depends on the site, compset, and configuration and for example if a "spinup" option is selected.

**--stdurbpt**

This option turns on the "stdurbpt\_pd" use-case for CLM\_NML\_USE\_CASE. This option can NOT be used for compsets that set the use-case to something besides present-day.

Lastly we discuss the dataset generation options. The dataset generation options are:

- aerdepgrid
- ndepgrid
- pftgrid
- soilgrid
- nopointdata
- owritesrfaer

The options that with a "grid" suffix all mean to create datasets using the global gridded information rather than using the site specific point data. By default the site specific point data is used. The "nopointdata" and "owritesrfaer" options have to do with file creation.

Because supported single-point datasets already have the data created for them, you MUST use the "nopointdata" and "ndepgrid" options when you are using a supported single-point site. You must use "ndepgrid" even for a compset without CN. You also can NOT use the options: "soilgrid", "pftgrid", "aerdepgrid", or "owritesrfaer".

**--aerdepgrid**

This option says to use the aerosol deposition files from the global dataset rather than creating an interpolated version.

This option must NOT be used when you are using a site that is a supported single point dataset.

**--ndepgrid**

This option says to use the Nitrogen deposition files from the global dataset rather than creating an interpolated version. This is only needed for compsets with CN.

**Note:** This option is *required* when you are using a site that is a supported single point dataset. This is true even when you are NOT using a compset with CN.

**--pftgrid**

This option says to use the PFT values provided on the global dataset rather than using the specific site based values from the `PTCLM_sitedata/*_pftdata.txt` file when creating the surface dataset.

This option must NOT be used when you are using a site that is a supported single point dataset.

--soilgrid

This option says to use the soil values provided on the global dataset rather than using the specific site based values from the `PTCLM_sitedata/*_soildata.txt` file when creating the surface dataset.

This option must NOT be used when you are using a site that is a supported single point dataset.

--nopointdata

This option says to NOT create any input datasets -- assume this step has already been done. If datasets weren't already created, your case will fail when you try to run it. In general the first time you run PTCLM for a new site you want it to generate new datasets, but the next time and future times you want to use this option so that it doesn't waste a lot of time rebuilding datasets over again.

**Note:** This option is *required* when you are using a site that is a supported single point dataset.

--owritesrfaer

This option says to overwrite any surface and/or aerosol deposition datasets that were already created. Otherwise, the creation of these files will be skipped if a file is already found (but it WILL create files if they don't exist).

This option must NOT be used when you are using a site that is a supported single point dataset.

**Note:** Note on the aerosol and Nitrogen deposition files. When the "aerdepgrid" and "ndepgrid" options are NOT used -- aerosol and Nitrogen deposition files will be created by interpolating from the global datasets. However, after these interpolated files are created you could customize them for your site with data that you provide. You could then write protect the files and use the "nopointdata" option so that PTCLM doesn't try to overwrite them in the future.

## Examples using PTCLM

Now let's give a few more complex examples using some of the options we have discussed above.

In this first example, we'll demonstrate using a supported single point dataset, which then requires using the "nopointdata" and "ndepgrid" options. We'll also demonstrate the compset option, "stdurbpt" and "caseidprefix" options.

**Example 6-2. Example of running PTCLM for the Mexicocity supported single point dataset**

```
> cd scripts/ccsm_utils/Tools/lnd/clm/PTCLM
> ./PTCLM.py -m bluefire -s 1x1_mexicocityMEX -d $CSMDATA --nopointdata --ndepgrid \
--stdurbpt -c I --caseidprefix `pwd`/myPTCLMcases/site
> cd myPTCLMcases/site_1x1_mexicocityMEX_I
> ./configure -case
# Now build and run normally
> ./site_1x1_mexicocityMEX_I.bluefire.build
# Here we show running interactively
> ./site_1x1_mexicocityMEX_I.bluefire.run
```

Now, let's demonstrate using a different group list, doing a spinup, running with Qian global forcing data, but using tower years to set the years to run over. This uses the options: sitegroupname, ad\_spinup, useQIAN, and QIANtower\_years.

**Example 6-3. Example of running PTCLM for a spinup simulation with Qian data for tower years.**

```
> cd scripts/ccsm_utils/Tools/lnd/clm/PTCLM
> ./PTCLM.py -m bluefire -s US-Hal -d $CSMDATA --sitegroupname AmeriFlux \
--ad_spinup --useQIAN --QIAN_tower_yrs
> cd ../../../../US-Hal_ICN_QIAN_ad_spinup
> ./configure -case
# Now build and run normally
> ./US-Hal_ICN_QIAN_ad_spinup.bluefire.build
# Here we show running interactively
> ./US-Hal_ICN_QIAN_ad_spinup.bluefire.run
```

Finally, let's demonstrate using a generic machine (which then requires the scratchroot option), using the global grid for PFT and soil types, and setting the run length to two months.

**Example 6-4. Example of running PTCLM on a generic machine with global PFT and soil types dataset**

```
> cd scripts/ccsm_utils/Tools/lnd/clm/PTCLM
# Note, see the the Section called Converting AmeriFlux Data for use by PTCLM with permission information
# to use the US-UMB data.
> ./PTCLM.py -m generic_darwin_intel -s US-UMB -d $CSMDATA --pftgrid --soilgrid \
--scratchroot $HOME --run_n 2 --run_units nmonths
> cd ../../../../US-UMB_ICN
> ./configure -case
# Now build
> ./US-UMB_ICN.generic_darwin_intel.build
# To get the files from the svn server...
# First list the files from the streams text file
> ./ccsm_utils/Tools/listfilesin_streams \
$HOME/US-UMB_ICN/run/clmPFT.1x1pt_US-UMB.stream.txt -l \
> Buildconf/datm.input_data_list
# And now run the script to export data to your machine
> ./ccsm_utils/Tools/check_input_data -export
# Here we show running interactively
> ./US-UMB_ICN.generic_darwin_intel.run
```

### **Caution**

Because of Bug 1364, when running this case as above we get a floating point error after reaching time-step 124 for the example exactly as above. Other machines or compilers probably won't have this problem. See the `models/lnd/clm/doc/KnownBugs (./KnownBugs)` file for more information on this problem.

### **Warning**

As documented in Bug 1368, spinning up the US-UMB site for a l2000CN compset gives zero Gross Primary Production (GPP). If the user wishes to use this site for CLMCN, they'll need to address this issue. See the `models/lnd/clm/doc/KnownBugs (./KnownBugs)` file for more information on this problem.

## **Adding data for use by PTCLM**

### **PTCLM Group Site Lists**

The "sitegroupname" option to PTCLM1.110726 looks for groups of sites in the files in the `PTCLM_sitedata` directory under the PTCLM directory. You can add new names available for this option including your own lists of sites, by adding more files in this directory. There are three files for each "sitegroupname": `$SITEGROUP_sitedata.txt`, `$SITEGROUP_soildata.txt` and `$SITEGROUP_pftdata.txt` (where `$SITEGROUP` is the name that would be entered as "sitegroupname" to PTCLM). Each file needs to have the same list of sites, but gives different information: site data, PFT data, and soil data respectively. Although the site codes need to be the same between the three files, the files do NOT have to be in the same order. Each file has a one-line header that lists the contents of each column which are separated by commas. The first column for each of the files is the "site\_code" which must be consistent between the three files. The site code can be any unique character string, but in general we use the AmeriFlux site code.

Site data file: `$SITEGROUP_sitedata.txt`): The header for this file is:

```
site_code,name,state,lon,lat,elev,startyear,endyear,alignyear
```

The columns: name, state, and elevation are informational only. Name is a longer descriptive name of the site, and state is the state for U.S. sites or country for non U.S. sites. The columns: lon and lat are the longitude and latitude of the location in decimal degrees. The last three columns are the start and ending year for the data and the align year for an 1850 case for the data. The align year is currently unused.

Soil data file: `$SITEGROUP_soildata.txt`): The header for this file is:

```
site_code,soil_depth,n_layers,layer_depth,layer_sand%,layer_clay%
```

The first three fields after "site\_code" are currently unused. The only two that are used are the percent sand and clay columns to set the soil texture.

PFT data file: \$SITEGROUP\_pftdata.txt): The header for this file is:

```
site_code,pft_f1,pft_c1,pft_f2,pft_c2,pft_f3,pft_c3,pft_f4,pft_c4,pft_f5,pft_c5
```

This file gives the vegetation coverage for the different vegetation types for the site. The file only supports up to five PFT's at the same time. The columns with "pft\_f" are the fractions for each PFT, and the columns with "pft\_c" is the integer index of the given PFT. Look at the pft-physiology file to see what the PFT index for each PFT type is.

## Dynamic Land-Use Change Files for use by PTCLM

There is a mechanism for giving site-specific land-use change in PTCLM. Adding site specific files to the PTCLM\_sitedata directory under PTCLM allows you to specify the change in vegetation and change in harvesting (for the CN model) for that site. Files are named: \$SITE\_dynpftdata.txt. There is a sample file for the US-Ha1 site called: US-Ha1\_dynpftdata.txt. The file has a one-line header with the information that the file has, and then one-line for each year with a transition. The header line is as follows:

```
trans_year,pft_f1,pft_c1,pft_f2,pft_c2,pft_f3,pft_c3,pft_f4,pft_c4,pft_f5,pft_c5,har_vh1,har_vh2, \
har_sh1,har_sh2,har_sh3,graze,hold_harv,hold_graze
```

This file only requires a line for each year where a transition or harvest happens. As in the "pftdata" file above "pft\_f" refers to the fraction and "pft\_c" refers to the PFT index, and only up to five vegetation types are allowed to co-exist. The last eight columns have to do with harvesting and grazing. The last two columns are whether to hold harvesting and/or grazing constant until the next transition year and will just be either 1 or 0. This file will be converted by the **PTCLM\_sitedata/cnvt\_trnsyrs2\_pftdynxtfile.pl** script in the PTCLM directory to a format that **mksurfdata** can read that has an entry for each year for the range of years valid for the compset in question.

## Converting AmeriFlux Data for use by PTCLM

AmeriFlux data comes in comma separated format and is available from:

<http://public.ornl.gov/ameriflux/dataproducts.shtml>

(<http://public.ornl.gov/ameriflux/dataproducts.shtml>). Before you download the data you need to agree to the usage terms.

Here is a copy of the usage terms from the web-site on June/13/2011.

"The AmeriFlux data provided on this site are freely available and were furnished by individual AmeriFlux scientists who encourage their use. Please kindly inform the appropriate AmeriFlux scientist(s) of how you are using the data and of any publication plans. Please acknowledge the data source as a citation or in the acknowledgments if the data are not yet published. If the AmeriFlux

Principal Investigators (PIs) feel that they should be acknowledged or offered participation as authors, they will let you know and we assume that an agreement on such matters will be reached before publishing and/or use of the data for publication. If your work directly competes with the PI's analysis they may ask that they have the opportunity to submit a manuscript before you submit one that uses unpublished data. In addition, when publishing, please acknowledge the agency that supported the research. Lastly, we kindly request that those publishing papers using AmeriFlux data provide preprints to the PIs providing the data and to the data archive at the Carbon Dioxide Information Analysis Center (CDIAC)."

The above agreement applies to the "US-UMB" dataset imported into our repository as well, and Gil Bohrer is the PI on record for that dataset.

The CESM can NOT handle missing data, so we recommend using the "Level 4" Gap filled datasets. The fields will also need to be renamed. The "WS" column becomes "WIND", "PREC" becomes "PRECmm", "RH" stays as "RH", "TA" becomes "TBOT", "Rg" becomes "FSDS", "Rgl" becomes "FLDS", "PRESS" becomes "PSRF". "ZBOT" can just be set to the constant of "30" (m). The units of Temperature need to be converted from "Celsius" to "Kelvin" (use the value in SHR\_CONST\_TKFRZ in the file models/csm\_share/shr/shr\_const.F90 of 273.15. The units of Pressure also need to be converted from "kPa" to "Pa". LATIXY, and LONGXY should also be set to the latitude and longitude of the site. With data hourly or half-hourly the default settings will be fine. But, if the temporal frequency is less than that see Example 5-7 for an example of how to change it.

## PTCLM transient example over a shorter time period

### Example 6-5. Example of running PTCLM for transient land-use 1991-2006 for US-Ha1 on bluefire

This is an example of using PTCLM for Harvard Forest (AmeriFlux site code US-Ha1). In order to do this we would've needed to have converted the AmeriFlux data into NetCDF format as show in the the Section called *Converting AmeriFlux Data for use by PTCLM* section above. Also note that this site has a site-specific dynamic land-use change file for it PTCLM\_sitedata/US-Ha1\_dynpftdata.txt in the PTCLM directory and this file will be used for land-use change and harvesting rather than the global dataset.

```
> cd scripts/ccsm_utils/Tools/lnd/clm/PTCLM
# We are going to use forcing data over 1991 to 2006, but we need to start with
# a transient compset to do so, so we use the 20th Century transient: 1850-2000
# Note: When creating the fpftdyn dataset for this site it will use the
#       PTCLM_sitedata/US-Ha1_dynpftdata.txt
# file for land-use change and harvesting
> ./PTCLM.py -n bluefire -s US-Ha1 -d $MYCSMDATA --sitegroupname AmeriFlux \
-c I_1850-2000_CN
> mkdir $MYCSMDATA/atm/datm7/CLM1PT_data/1x1pt_US-Ha1
> cd $MYCSMDATA/atm/datm7/CLM1PT_data/1x1pt_US-Ha1
# Copy data in NetCDF format to this directory, filenames should be YYYY-MM.nc
# The fieldnames on the file should be:
#   FLDS,FSDS,LATIXY,    PRECTmm,PSRF,RH,TBOT,WIND,ZBOT
# With units
#   W/m2,W/m2,degrees_N,degrees_E,mm/s,    Pa,    %,    K,    m/s,    m
# The time coordinate units should be: days since YYYY-MM-DD 00:00:00
> cd ../../US-Ha1_I_1850-2000_CN
# We need to turn cold-start on, so it doesn't expect an initial condition file
# preferably, you would generate your own initial condition file and then use
# the --finidat option to PTCLM to enter it
> ./xmlchange -file env_conf.xml -id CLM_FORCE_COLDSTART -val on
# Now we need to set the start date to 1991, and have it cycle forcing data
# over 1991 to 2006
```

```

> ./xmlchange -file env_conf.xml -id RUN_STARTDATE -val 1991-01-01
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_ALIGN -val 1991
> ./xmlchange -file env_conf.xml -id DATM_CLMNCEP_YR_END -val 2006
> ./xmlchange -file env_conf.xml -id CLM_NAMELIST_OPTS -val \
# Similarly for Nitrogen deposition data we cycle over: 1991 to 2006
"model_year_align_ndep=1991,stream_year_first_ndep=1991,stream_year_last_ndep=2006"
# Now configure the case, and we'll edit the DATM namelist for prescribed aerosols
> ./configure -case
# We also need to change the DATM to run with aerosols over the 1991-2006 period
cat << EOF > patch.diff
*** datm.buildnml.csh.orig 2011-06-14 09:28:20.000000000 -0600
--- datm.buildnml.csh 2011-06-14 09:28:57.000000000 -0600
*****
*** 32,38 ****
    dataMode      = 'CLMNCEP'
    domainfile   = '$DOMAINFILE'
    streams      = 'clm1PT.1x1pt_US-Ha1.stream.txt 1991 1991 2006 ',
!           'presaero.stream.txt 1849 1849 2006'
    vectors      = 'null','null'
    mapmask      = 'nomask','nomask'
    mapalgo      = 'nn','nn'
--- 32,38 ----
    dataMode      = 'CLMNCEP'
    domainfile   = '$DOMAINFILE'
    streams      = 'clm1PT.1x1pt_US-Ha1.stream.txt 1991 1991 2006 ',
!           'presaero.stream.txt 1991 1991 2006'
    vectors      = 'null','null'
    mapmask      = 'nomask','nomask'
    mapalgo      = 'nn','nn'
EOF
# Apply the above patch to the DATM build namelist file
> patch Buildconf/datm.buildnml.csh patch.diff

```

## A bit about the structure of PTCLM, what it does, and how it works

A large part of PTCLM just sets up the different options and does error checking on the options given. PTCLM then uses the options provided to use **create\_newcase** to create a new case. It then queries both the case directory and/or the XML database (using `queryDefaultNamelist.pl` in `models/lnd/clm/bld`) and does other settings for the case. It then runs the different CLM tools in turn to create the necessary datasets and points to them in the case with the `CLM_USRDAT_NAME` option. It runs **mkgriddata**, **mksurfdata.pl**, and **mkdatadomain** as well as the **aerdepregrid.ncl** and **ndepreregrid.ncl** NCL scripts. **mkgriddata** and **mkdatadomain** have template namelist files in the `scripts/ccsm_utils/Tools/lnd/clm/PTCLM/usr_files` directory. When running **mksurfdata.pl** if it finds a `$SITE_dynpftdata.txt` in the `scripts/ccsm_utils/Tools/lnd/clm/PTCLM/PTCLM_sitedata` directory it will use that file for transient landuse changes (there's a sample file for "US-Ha1" called `US-Ha1_dynpftdata.txt`). It modifies the different `env*.xml` using **xmlchange** and creates an initial `user_nl_clm` filename. After PTCLM is run you can then make changes to the case by hand, and configure, build and run as normal.

There is a simple test script to test PTCLM. See the Section called *Testing PTCLM use the PTCLM test script* in Appendix C for more information on using it.

# Chapter 7. Trouble Shooting Problems

In this chapter we give some guidance on what to do when you encounter some of the most common problems. We can't cover all the problems that a user could potentially have, but we will try to help you recognize some of the most common situations. And we'll give you some suggestions on how to approach the problem to come up with a solution.

In general you will run into one of three type of problems:

1. *configure-time*
2. *build-time*
3. *run-time*

You may also run into problems with **create\_newcase** itself, or with the archiving scripts -- for those problems you should consult the CESM1.0.4 Scripts User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>).

## Trouble with Configuration

The first type of problem happens when you invoke the **configure -case** command. This indicates there is something wrong with your template files, or input datasets, or the details of what you are trying to configure the model to do. There's also a trouble-shooting chapter in the CESM1.0.4 Scripts User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>). Many of the problems with configuration can be resolved with the guidelines given there. Here we will restrict ourselves to problems from the CLM or DATM templates or input files.

### Example 7-1. Example of configure problem with missing datasets

```
> ./create_newcase -case T31rcp6 -res T31_g37 -compset IRCP60CN \
-mach bluefire
> ./configure -case
```

The following is what is displayed to the screen.

```
Generating resolved namelist, prestage, and build scripts
configure done.
adding use_case 1850-2100_rcp6_transient defaults for var clm_demand with val fpftdyn
adding use_case 1850-2100_rcp6_transient defaults for var clm_start_type with val startup
adding use_case 1850-2100_rcp6_transient defaults for var model_year_align_ndep with val
1850
adding use_case 1850-2100_rcp6_transient defaults for var rcp with val 6
adding use_case 1850-2100_rcp6_transient defaults for var sim_year with val 1850
adding use_case 1850-2100_rcp6_transient defaults for var sim_year_range with val
1850-2100
adding use_case 1850-2100_rcp6_transient defaults for var stream_year_first_ndep with val
1850
adding use_case 1850-2100_rcp6_transient defaults for var stream_year_last_ndep with val
2100
adding use_case 1850-2100_rcp6_transient defaults for var use_case_desc with val Simulate
transient land-use, aerosol and Nitrogen deposition changes with historical data from
1850 to 2005 and then with the RCP6 scenario from AIM

build-namelist - No default value found for fpftdyn.
Are defaults provided for this resolution and land mask?
ERROR: generate_resolved.csh error for lnd template
```

```
configure error: configure generated error in attempting to created resolved scripts
```

The important thing to note here is the line:

```
ERROR: generate_resolved.csh error for lnd template
```

which tells us that the problem is in the land template. It may also indicate problems in one of the other templates (atm, ccsim, cpl, glc, ice, or ocn), in which case you should consult the appropriate model user's guide, and examine the given template file in `Tools/Templates`. For more information on working with template files see Appendix A.

In the example above, it's obvious that the problem is coming from the CLM **build-namelist**, in other situations it might not be so obvious where the problem is occurring. In such cases it might be useful to add a "set echo" command to the top of the template file so that each command in the template will be echoed to the screen and you can see what is happening and where the error is occurring.

```
set echo
```

In the example, the error is that the CLM XML database does NOT have a `fpftdyn` for the given resolution, rcp scenario and ocean mask. That means you will need to create the file and then supply the file into your case. See Chapter 2 for more information on creating files, and see Chapter 3 for more information on adding files to the XML database. Alternatively, you can provide the file to your case by creating a user namelist as shown in the Section called *Configure time User Namelist* in Chapter 1.

**Note:** The two most common problems from your CLM template will be errors from the CLM **configure** or **build-namelist**. For more information on these scripts see: the Section called *More information on the CLM **configure** script* in Chapter 1 and the section on `CLM_BLDNML_OPTS`.

## Trouble with Building

Here's an example of running the build for a case and having it fail in the land model build. As you can see it lists which model component is being built and the build log for that component.

```
CCSM BUILDEXE SCRIPT STARTING
- Build Libraries: mct pio csm_share
Sat Jun 19 21:21:19 MDT 2010 /ptmp/erik/test_build/mct/mct.bldlog.100619-212107
Sat Jun 19 21:22:18 MDT 2010 /ptmp/erik/test_build/pio/pio.bldlog.100619-212107
Sat Jun 19 21:23:18 MDT 2010
/ptmp/erik/test_build/csm_share/csm_share.bldlog.100619-212107
Sat Jun 19 21:24:00 MDT 2010 /ptmp/erik/test_build/run/cpl.bldlog.100619-212107
Sat Jun 19 21:24:00 MDT 2010 /ptmp/erik/test_build/run/atm.bldlog.100619-212107
Sat Jun 19 21:24:06 MDT 2010 /ptmp/erik/test_build/run/lnd.bldlog.100619-212107
ERROR: clm.buildexe.csh failed, see /ptmp/erik/test_build/run/lnd.bldlog.100619-212107
ERROR: cat /ptmp/erik/test_build/run/lnd.bldlog.100619-212107
```

You can then examine the build log that failed and see what went wrong. Most compilers will give the full filepath and line number for the file that failed to compile.

## Trouble with Running

Tracking down problems while the model is running is much more difficult to do than configure or build problems. In this section we will give some suggestions on how to find run time problems. Below we show the log file results of a job that aborted while running.

```
CCSM PRESTAGE SCRIPT HAS FINISHED SUCCESSFULLY
Sun Jun 20 18:24:06 MDT 2010 -- CSM EXECUTION BEGINS HERE
Sun Jun 20 18:24:35 MDT 2010 -- CSM EXECUTION HAS FINISHED
Model did not complete - see /ptmp/erik/test_run/run/cpl.log.100620-182358
```

In the next section we will talk about using the different log files to track down problems, and find out where the problem is coming from. In the section after that we give some general advice on debugging problems and some suggestions on ideas that may be helpful to track the problem down. Some of the examples below are from the `models/lnd/clm/doc/KnownBugs (./KnownBugs)` file.

## Tracking Problems by Querying Log Files

The first thing to do when tracking down problems is to query the different log files to see if you can discover where the problem occurs, and any error messages about it. It's important to figure out if the problem comes in at initialization or in the run phase of the model, and in which model component the problem happens. There are different log files for the different major components, and they all end with the date and time in YYMMDD-HHMMSS format (2-digit: year, month, day, hour minute and second). When the model runs to completion the log files will be copied to the `logs` directory in the script directory, but when the model fails they will remain in the `run` directory. Here's an example list of log files from an "I" case where the model dies in the land model initialization. For "I" cases the sea-ice and ocean components are just stubs and don't create log files (and unless running with the active land-ice model "glc" log files won't be created either).

```
atm.log.100620-182358
ccsm.log.100620-182358
cpl.log.100620-182358
lnd.log.100620-182358
```

## The coupler log file

The first log file to check is the coupler log file so that you can see where the model dies and which model component it fails in. When the model dies at initialization the last model component listed is the component that failed.

Example of a case that fails in the CLM land model initialization.

```
(seq_tiemgr_clockPrint)      Prev Time    = 00001201    00000
```

```
(seq_timemgr_clockPrint)      Next Time    = 99991201   00000
(seq_timemgr_clockPrint)      Interval yms =      9999         0          0

(seq_mct_drv) : Initialize each component: atm, lnd, ocn, and ice
(seq_mct_drv) : Initialize atm component
(seq_mct_drv) : Initialize lnd component
```

## The CCSM log file

The CCSM log files are to some extent the "garbage collection" of log output. The CLM sends its output from its master processor, but sends other output and possibly errors to the CCSM log file. Because of this, often error messages are somewhere in the CCSM log file. However, since there is so much other output it may be difficult to find. For example, here is some output from an older version of CESM (CESM1.0.2) where the RTM river routing file (before it was converted to NetCDF) was not provided and the error on the open statement for the file was embedded near the end of the CCSM log file.

```
NODE# NAME
(     0) be1105en.ucar.edu
"/gpfs/proj2/fis/cgd/home/erik/clm_trunk/models/lnd/clm/src/riverroute/RtmMod.F90", line
239: 1525-155 The file name provided in the OPEN statement for unit 1 has zero length or
contains all blanks. The program will recover by ignoring the OPEN statement.
"/gpfs/proj2/fis/cgd/home/erik/clm_trunk/models/lnd/clm/src/riverroute/RtmMod.F90", line
241: 1525-001 The READ statement on the file fort.1 cannot be completed because the end
of the file was reached. The program will stop.

Running: ./ccsm.exe
Please wait...

Memory usage for ./ccsm.exe (task # 0) is:      51696 KB. Exit status: 1. Signal: 0
```

Although the example is from an earlier version of the model it still serves to illustrate finding problems from the CCSM log file.

When working with the CCSM log file, for a run-time problem, you will need to be able to separate its output into three categories: pre-crash, crash, and post-crash. The pre-crash section is everything that is normal output for good operation of the model. The crash section is the section where the model dies and reports on the actual problem. the post-crash section is the cleanup and finalization after the model dies. The most important part of this of course is the crash section. The tricky part is distinguishing it from the other sections. Also because the CCSM log file most likely has duplicated output from multiple processors it is even more difficult to distinguish the different sections and to some extent the sections may be intertwined, as different processors reach the different sections at different times. Because, of this reducing the number of processors for your simulation may help you sort out the output in the file (see the Section called *Run with a smaller set of processors*). Also much of the output from the CCSM log file are system level information having to do with MPI multiprocessing. Usually you can ignore this information, but it makes it more difficult to trudge through.

Sometimes the CCSM log file is the ONLY file available, because the model terminates early in initialization. In this case understanding the output in the CCSM log file becomes even more important. This also indicates the model did NOT advance far enough to reach the initialization of the individual model components. This may mean that the initialization of the multiprocessing for MPI and/or OpenMP failed, or that the reading of the driver namelist file "drv\_in" failed.

Here we show those three sections for a ccsm log file where a two task job failed on reading the namelist file. For a typical job with many tasks similar sections of this will be repeated not just twice but for each task and hence make it harder to read.

#### *Pre-crash section of the ccsm log file*

```

ATTENTION: 0031-386 MP_INSTANCES setting ignored when LoadLeveler is not being used.
ATTENTION: 0031-386 MP_INSTANCES setting ignored when LoadLeveler is not being used.
ATTENTION: 0031-378 MP_EUIDEVICE setting ignored when LoadLeveler is not being used.
ATTENTION: 0031-386 MP_INSTANCES setting ignored when LoadLeveler is not being used.
0:INFO: 0031-724 Executing program: </usr/local/lsf/7.0/aix5-64/bin/lsnrt_run>
1:INFO: 0031-724 Executing program: </usr/local/lsf/7.0/aix5-64/bin/lsnrt_run>
0:/contrib/bin/ccsm_launch: process 401894 bound to logical CPU 0 on host be0310en.ucar.edu ...
1:/contrib/bin/ccsm_launch: process 439264 bound to logical CPU 1 on host be0310en.ucar.edu ...
0:INFO: 0031-619 64bit(us, Packet striping on) ppe_rmas MPC1_MSG: MPI/MPC1 library was compiled on   Wed Aug  5 13:36:06 2009
0:
1:LAPI version #14.26 2008/11/23 11:02:30 1.296 src/rsct/lapi/lapi.c, lapi, rsct_rpt53, rpt53s004a 09/04/29 64bit(us)  library comp...
1:.
1:LAPI is using lightweight lock.
0:LAPI version #14.26 2008/11/23 11:02:30 1.296 src/rsct/lapi/lapi.c, lapi, rsct_rpt53, rpt53s004a 09/04/29 64bit(us)  library comp...
0:.
0:LAPI is using lightweight lock.
0:Use health ping for failover/recovery
1:Use health ping for failover/recovery
0:Initial communication over instance 2.
1:Initial communication over instance 0.
1:IB RDMA initialization completed successfully
1:The MPI shared memory protocol is used for the job
0:IB RDMA initialization completed successfully
0:LAPI job ID for this job is: 1684890719
0:The MPI shared memory protocol is used for the job
0:(seq_comm_setcomm) initialize ID ( 7 GLOBAL ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_setcomm) initialize ID ( 2 ATM ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_setcomm) initialize ID ( 1 LND ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_setcomm) initialize ID ( 4 ICE ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_setcomm) initialize ID ( 5 GLC ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_setcomm) initialize ID ( 3 OCN ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_setcomm) initialize ID ( 6 CPL ) pelist = 0 1 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_joincomm) initialize ID ( 8 CPLATM ) join IDs = 6 2 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_joincomm) initialize ID ( 9 CPLLND ) join IDs = 6 1 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_joincomm) initialize ID ( 10 CPLICE ) join IDs = 6 4 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_joincomm) initialize ID ( 11 CPLOCN ) join IDs = 6 3 ( npes = 2 ) ( nthreads = 1 )
0:(seq_comm_joincomm) initialize ID ( 12 CPLGLC ) join IDs = 6 5 ( npes = 2 ) ( nthreads = 1 )
0:
0: (seq_comm_printcomms) ID layout : global pes vs local pe for each ID
0:      gpe      LND      ATM      OCN      ICE      GLC      CPL      GLOBAL      CPLATM      CPLLND      CPLICE      CPLOCN      CPLGLC      nthreads
0:      ---      ---      ---      ---      ---      ---      ---      ---      ---      ---      ---      ---      ---      ---      ---
0:      0 :      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1
1:      1 :      1      1      1      1      1      1      1      1      1      1      1      1      1      1      1
1:.
0: (t_inif) Read in prof_inparm namelist from: drv_in
1: (seq_io_init) cpl_io_stride, iotasks or root out of bounds - resetting to defaults 4 0 1
0: piolib_mod.f90 1353 1 2 1 2
1: piolib_mod.f90 1353 1 2 1 2
0: pio_support::pio_die:: myrank= 0 : ERROR: piolib_mod.f90: 1354 : not enough procs for the stride
1: pio_support::pio_die:: myrank= 1 : ERROR: piolib_mod.f90: 1354 : not enough procs for the stride

```

#### *Crash section of the ccsm log file*

```

0:
0: Traceback:
1:
1: Traceback:
0:   Offset 0x00000c4c in procedure __pio_support_NMOD_piodie, near line 88 in file pio_support.F90.in
1:   Offset 0x00000c4c in procedure __pio_support_NMOD_piodie, near line 88 in file pio_support.F90.in
0:   Offset 0x00000fd0 in procedure __piolib_mod_NMOD_init, near line 1354 in file piolib_mod.F90
1:   Offset 0x00000fd0 in procedure __piolib_mod_NMOD_init, near line 1354 in file piolib_mod.F90
1:   Offset 0x00000398 in procedure __seq_io_mod_NMOD_seq_io_init, near line 247 in file /gpfs/proj2/fis/cgd/home/erik/clm_trunk/r...
0:   Offset 0x00000398 in procedure __seq_io_mod_NMOD_seq_io_init, near line 247 in file /gpfs/proj2/fis/cgd/home/erik/clm_trunk/r...
0:   Offset 0x0001aa88 in procedure ccsm_driver, near line 465 in file /gpfs/proj2/fis/cgd/home/erik/clm_trunk/models/driv...

```

```

0:    --- End of call chain ---
1:    Offset 0x0001aa88 in procedure ccsm_driver, near line 465 in file /gpfs/proj2/fis/cgd/home/erik/clm_trunk/models/drv/driver/
1:    --- End of call chain ---

```

### *Post-crash section of the ccsm log file*

```

1:Communication statistics of task 1 is associated with task key: 1684890719_1
0:Communication statistics of task 0 is associated with task key: 1684890719_0
0:
0:Running: ./ccsm.exe
0:Please wait...
0:
0:Memory usage for ./ccsm.exe (task # 0) is: 198892 KB. Exit status: 134. Signal: 0
1:
1:Running: ./ccsm.exe
1:Please wait...
1:
1:Memory usage for ./ccsm.exe (task # 0) is: 198572 KB. Exit status: 134. Signal: 0
INFO: 0031-656 I/O file STDOUT closed by task 0
INFO: 0031-656 I/O file STDERR closed by task 0
ERROR: 0031-250 task 0: IOT/Abort trap
INFO: 0031-656 I/O file STDOUT closed by task 1
INFO: 0031-656 I/O file STDERR closed by task 1
ERROR: 0031-250 task 1: IOT/Abort trap
INFO: 0031-639 Exit status from pm_respond = 0
ATTENTION: 0031-386 MP_INSTANCES setting ignored when LoadLeveler is not being used.
Job /usr/local/lsf/7.0/aix5-64/bin/poejob /contrib/bin/ccsm_launch /contrib/bin/job_memusage.exe ./ccsm.exe

TID      HOST_NAME      COMMAND_LINE          STATUS           TERMINATION_TIME
=====  ======  ======  ======  ======
00000  be0310en  /contrib/bin/ccs  Exit (134)        08/31/2010 12:32:57
00001  be0310en  /contrib/bin/ccs  Exit (134)        08/31/2010 12:32:57

```

## The CLM log file

Of course when you are working with and making changes to CLM, most of your focus will be on the CLM log file and the errors it shows. As already pointed out if you don't see errors in the `lnd.log.*` file you should look in the `ccsm.log.*` to see if any errors showed up there.

Here's an example of the `lnd.log.*` file when running PTS\_MODE with initial conditions (this is bug 1025 in the `models/lnd/clm/doc/KnownLimitations (./KnownLimitations)` file).

```

Successfully initialized variables for accumulation

reading restart file I2000CN_f09_g16_c100503.clm2.r.0001-01-01-00000.nc
Reading restart dataset
ERROR - setlatlon.F:Cant get variable dim for lat or lsmlat
ENDRUN: called without a message string

```

## The DATM log file

When working with "I cases" the second most common problems after CLM problems are problems with the data atmosphere model. So examining the `atm.log.*` is important.

Here's an example of a problem that occurs when the wrong prescribed aerosol file is given to a `pt1_pt1` simulation.

```
(datm_comp_init) atm mode = CLMNCEP
(shr_strdata_init) calling shr_dmodel_mapSet for fill
(shr_strdata_init) calling shr_dmodel_mapSet for remap
('shr_map_getWts') ERROR: yd outside bounds 19.5000000000000000
(shr_sys_abort) ERROR: ('shr_map_getWts') ERROR yd outside 90 degree bounds
(shr_sys_abort) WARNING: calling shr_mpi_abort() and stopping
```

## The batch log files

The names of the batch log files will depend on the batch system of the machine that is being used. They will normally be in the script directory. Usually, they don't contain important information, but they are a last resort place to look for error messages. On the NCAR IBM system "bluefire" the batch files are called with names that start with "poe" and then either "stderr" or "stdout", with the job number at the end.

## General Advice on Debugging Run time Problems

Here are some suggestions on how to track down a problem while running. In general if the problem still occurs for a simpler case, it will be easier to track down.

1. *Run in DEBUG mode*
2. *Run with a smaller set of processors*
3. *Run in serial mode with a single processor*
4. *Run at a lower resolution*
5. *Run a simpler case*
6. *Run with a debugger*

## Run in DEBUG mode

The first thing to try is to run in DEBUG mode so that float point trapping will be triggered as well as array bounds checking and other things the compiler can turn on to help you find problems. To do this edit the `env_build.xml` file and set `DEBUG` to `TRUE` as follows:

```
> ./xmlchange -file env_build.xml -id DEBUG -val TRUE
```

## Run with a smaller set of processors

Another way to simplify the system is to run with a smaller set of processors. You will need to clean the configure and edit the `env_mach_pes.xml`. For example, to run with four processors:

```
> ./configure -cleanall
> ./xmlchange -file env_mach_pes.xml -id NTASKS_ATM -val 4
> ./xmlchange -file env_mach_pes.xml -id NTASKS_LND -val 4
> ./xmlchange -file env_mach_pes.xml -id NTASKS_ICE -val 4
> ./xmlchange -file env_mach_pes.xml -id NTASKS_OCN -val 4
> ./xmlchange -file env_mach_pes.xml -id NTASKS_CPL -val 4
> ./xmlchange -file env_mach_pes.xml -id NTASKS_GLC -val 4
> ./configure -case
```

Another recommended simplification is to run without threading, so set the NTHRDS for each component to "1" if it isn't already. Sometimes, multiprocessing problems require a certain number of processors before they occur so you may not be able to debug the problem without enough processors. But, it's always good to reduce it to as low a number as possible to make it simpler. For threading problems you may have to have threading enabled to find the problem, but you can run with 1, 2, or 3 threads to see what happens.

## Run in serial mode with a single processor

Simplifying to one processor removes all multi-processing problems and makes the case as simple as possible. If you can enable `USE_MPI_SERIAL` you will also be able to run interactively rather than having to submit to a job queue, which sometimes makes it easier to run and debug. If you can use `USE_MPI_SERIAL` you can also use threading, but still run interactively in order to use more processors to make it faster if needed.

```
> ./configure -cleanall
# Set tasks and threads for each component to 1
# You could also set threads to something > 1 for speed, but still
# run interactively if threading isn't an issue.
> ./xmlchange -file env_mach_pes.xml -id NTASKS_ATM -val 1
> ./xmlchange -file env_mach_pes.xml -id NTHRDS_ATM -val 1
> ./xmlchange -file env_mach_pes.xml -id NTASKS_LND -val 1
> ./xmlchange -file env_mach_pes.xml -id NTHRDS_LND -val 1
> ./xmlchange -file env_mach_pes.xml -id NTASKS_ICE -val 1
> ./xmlchange -file env_mach_pes.xml -id NTHRDS_ICE -val 1
> ./xmlchange -file env_mach_pes.xml -id NTASKS_OCN -val 1
> ./xmlchange -file env_mach_pes.xml -id NTHRDS_OCN -val 1
> ./xmlchange -file env_mach_pes.xml -id NTASKS_CPL -val 1
> ./xmlchange -file env_mach_pes.xml -id NTHRDS_CPL -val 1
> ./xmlchange -file env_mach_pes.xml -id NTASKS_GLC -val 1
> ./xmlchange -file env_mach_pes.xml -id NTHRDS_GLC -val 1
# If mpi-serial capability is available on the machine you are using
# set USE_MPI_SERIAL to true so that you can run interactively
> ./xmlchange -file env_conf.xml -id USE_MPI_SERIAL -val TRUE
> ./configure -case
# Then build your case
# And finally run, by running the *.run script interactively
# (If you were able to set USE_MPI_SERIAL to true)
```

## **Run at a lower resolution**

If you can create a new case running at a lower resolution and replicate the problem it may be easier to solve. This of course requires creating a whole new case, and trying out different lower resolutions.

## **Run a simpler case**

Along the same lines, you might try running a simpler case, trying another compset with a simpler setup and see if you can replicate the problem and then debug from that simpler case. Again, of course you will need to create new cases to do this.

## **Run with a debugger**

Another suggestion is to run the model with a debugger such as: **dbx**, **gdb**, or **totalview**. Often to run with a debugger you will need to reduce the number of processors as outlined above. Some debuggers such as **dbx** will only work with one processor, while more advanced debuggers such as **totalview** can work with both MPI tasks and OMP threads. Even simple debuggers though can be used to query core files, to see where the code was at when it died (for example using the **where** in **dbx** for a core file can be very helpful. For help in running with a debugger you will need to contact your system administrators for the machine you are running on.

# Appendix A. Editing Template Files Before Configure

The last kind of customization that you can do for a case, before configure is run is to edit the templates. The CLM template is in `models/lnd/clm/bld/clm.cpl7.template`, the DATM template is in `models/atm/datm/bld/datm.cpl7.template`, and the driver templates are in the `models/drive/bld` directory and are named: `ccsm.template` and `cpl.template`. When a case is created they are also copied to the `Tools/Templates` directory underneath your case. If you want to make changes that will impact all your cases, you should edit the template files under the `models` directory, but if you want to make a change ONLY for a particular case you should edit the template under that specific case.

**Note:** Editing the template files is NOT for the faint of heart! We recommend this ONLY for experts! It's difficult to do because the template is a script that actually creates another script. So part of the script is echoing the script to be created and part of it is a script that is run when "configure -case" is run. As a result any variables in the part of the script that is being echoed have to be escaped like this:

```
\$VARIABLE
```

But, in other parts of the script that is run, you can NOT escape variables. So you need to understand if you are in a part of the script that is echoing the script to be created, or in the part of the script that is actually run.

If you can customize your case using: compsets, `env_*.xml` variables, or a user namelist, as outlined in Chapter 1 you should do so. The main reason to actually edit the template files, is if you are in a situation where the template aborts when you try it run it when "configure -case" is run. The other reason to edit the template is if you are CLM developer and need to make adjustments to the template because of code or script updates. An example of modifying the DATM template is in Example 4-9 where sed is used to modify the path for `CPLHIST3HrWx` data.

## Outline of the CLM template

The outline of the CLM template is as follows:

```
# set up options for clm configure and then run clm configure
$SCDEROOT/lnd/clm*/bld/configure <options>
# set up options for clm build-namelist and then run clm build-namelist
$SCDEROOT/lnd/clm*/bld/build-namelist <options>
# echo the $CASEBUILD/clm.buildnml.csh script out
cat >! $CASEBUILD/clm.buildnml.csh << EOF1
# NOTE: variables in this section must be escaped
EOF1
# Remove temporary namelist files

# echo the $CASEBUILD/clm.buildexe.csh script out
cat > $CASEBUILD/clm.buildexe.csh <<EOF2
# NOTE: variables in this section must be escaped
EOF2
# Remove temporary configure files
```

## Outline of the DATM template

The outline of the DATM template is as follows:

```

# Check $GRID to set the $DOMAINFILE and $DOMAINPATH

# Check DATM_PRESAERO to set the prescribed aerosol option
# If CLM_USRDAT_NAME is set and $DOMAINFILE is NOT -- set it by CLM_USRDAT_NAME
# Ensure $DOMAINFILE is set or else abort

=====
# Create resolved prestage data script
=====
cat >! $CASEBUILD/datm.buildnml.csh << EOF1
# NOTE: variables in this section must be escaped
EOF1
# Major if blocks look at DATM_MODE:
# the if blocks setup streams and run Tools/build_streams to create stream files
----- CLM_QIAN mode -----
else if ($DATM_MODE == "CLM_QIAN" ) then
.
    # Customize CLM_QIAN options here

    # A.) Setup datm_atm_in namelist
    cat >! $CASEBUILD/datm.buildnml.csh << EOF
    cat >! datm_atm_in << EOF1
    # NOTE: variables in this section must be escaped
    EOF1
    EOF

    # B.) Setup options to build_streams
.
.
.

----- CLM1PT    mode -----
else if ($DATM_MODE == "CLM1PT" ) then
.
    # Customize CLM1PT options here

    # A.) Setup datm_atm_in namelist
    cat >! $CASEBUILD/datm.buildnml.csh << EOF
    cat >! datm_atm_in << EOF1
    # NOTE: variables in this section must be escaped
    EOF1
    EOF

    # B.) Setup options to build_streams
.
.
.

----- CPLHIST 3-hourly time-averaging mode
-----
else if ($DATM_MODE == "CPLHIST3HrWx" ) then
.
    # Customize CPLHIST3HrWx options here

    # A.) Setup datm_atm_in namelist
    cat >! $CASEBUILD/datm.buildnml.csh << EOF
    cat >! datm_atm_in << EOF1
    # NOTE: variables in this section must be escaped
    EOF1
    EOF

    # B.) Setup options to build_streams
.
.
.
```

```

.
.
.
#----- INVALID mode -----else
echo "ERROR: unrecognized DATM_MODE = \$DATM_MODE "
exit -1
endif

=====
# Create prescribed aero streams if appropriate
=====

.
.

=====
# Create remaining resolved namelist
=====

cat >! $CASEBUILD/datm.buildnml.csh << EOF
cat >! datm_in << EOF1
# NOTE: variables in this section must be escaped
.
.
EOF1

EOF

=====
# Create script to build executable
=====

cat > $CASEBUILD/datm.buildexe.csh <<EOF
#!/bin/csh -f
# NOTE: variables in this section must be escaped
EOF

=====
# end of script
=====

```

## Adding a new DATM\_MODE to the DATM template

### The steps to adding a new DATM\_MODE

1. Add a new "if" block to the DATM template

As you can see from the Section called *Outline of the DATM template* above there are major "if" blocks for the different DATM\_MODE's. So adding a new DATM\_MODE means adding a new "if" block. The two major parts of each DATM\_MODE block are:

Setup `datm_atm_in` namelist  
 Setup options to **build\_streams**

2. In the "if" block create the `datm_atm_in` namelist

See the Section called *Customizing the DATM Namelist and Streams files* in Chapter 1 for some notes about the DATM namelist and streams files. That and the DATM User's Guide ([http://www.cesm.ucar.edu/models/cesm1.0/data8/data8\\_doc/book1.html](http://www.cesm.ucar.edu/models/cesm1.0/data8/data8_doc/book1.html)) should give you guidance on how to setup the namelist for your case.

3. In the "if" block create options to and call **build\_streams**

The next part of the "if" block in the DATM template file to work with is the call to **build\_streams**. You may need to add additional options to it. You may also need to call it multiple times for multiple streams. You will also likely need to add a new source option to it with the "-s" option. For more information on **build\_streams** do the following.

### Example A-1. Getting help with build\_streams for DATM

```
> scripts/ccsm_utils/Tools/build_streams -help
```

The output of the above command is:

```
SYNOPSIS
    build_streams [options]

REQUIRED OPTIONS:

    -datasource "source"  [or -s] Use given data source type.
    AND Either
        -model      "name"      [or -m] Give the model type name (cice datm dlnd dice docn) and \
use the appropriate model template file.
                                                use the template file named
                                                (.../.../.../.../scripts/ccsm_utils/Tools/..Components/ \
name.template.streams.xml)
    Or
        -template "filename"   [or -t] Full pathname to the input streams template file to use.

OPTIONS THAT MAY BE REQUIRED (depending on input template and datasource)

    -case      "casename"   [or -c] Replace any template case indicators (%c see below) with this \
casename.
    -domain    "filename"   [or -do] Use given domain filename for domain
                                (required when template uses %do).
    -dompath   "directory"  [or -dp] Use given domain filepath for domain files.
                                (required when template uses %dp).
    -filepath  "directory"  [or -p] Full path to where datafiles are (required when template uses \
%p).
                                (Default use path in template file relative to -csmdata \
option)
    -res       "resolution" [or -r] Resolution for filenames if source in template is resolution \
dependent.
    -yearfirst "year"      [or -b] The first year to cycle input datasets over.
                                (required input if input template has year-month indicators \
(%y or %ym) )
    -yearlast  "year"      [or -e] The last year to cycle input datasets over.
                                (required input if input template has year-month indicators \
(%y or %ym) )

OPTIONS

    -csmdata  "directory"   Give input ccsm inputdata directory. (replaces %d used in \
template) (default $DIN_LOC_ROOT)
    -fexists   streams
                                Check that the domain and filenames indicated by the output \
file actually exist.
    -filenames "file_or_indicator" Use input fileNames string for filenames.
                                (Replace following strings appropriately: )
                                %c = Case (from above -case command line option)
                                %do = Use domain file
                                %y = Year (through range given from begyear to endyear)
    \
                                %ym = Year-Month (all 12 months through year range)
                                %6ym = Like %ym but 6 digit year (ie. YYYYYY-MM).
                                (can replace the 6 with any digit 1-9)

    -help
    -lastmonth
    -outfile   "filename"   [or -o] Output streams file to create. (default send to stdout)
    -verbose   [or -v] Turns on verbose printing mode.
```

#### 4. Add new streams templates to the DATM datm.template.streams.xml file

As part of modifying the behavior of **build\_streams** you will also have to edit the models/atm/datm7/bld/datm.template.streams.xml file as well (or the local version in your \$CASENAME/Tools/Templates directory for a particular case). The template is an XML file much like the output streams file, but there are attributes to distinguish which fields will be used based on things like: RESOLUTION or datasource. And there are filename indicators (starting with a "%") that get translated into various things such as:

```
%c = Case (from above -case command line option)
%do = Use domain file
%y = Year (through range given from begyear to endyear)
%ym = Year-Month (all 12 months through year range)
%6ym = Like %ym but 6 digit year (ie. %YYYYYY-MM). (can replace the 6 with any digit 1-9)
```

5. Add a new valid\_value to the config\_definition.xml file in scripts.

Adding a new DATM\_MODE also requires adding a new valid\_value to scripts/ccsm\_utils/Case.template/config\_definition.xml. This enables the scripts to recognize the new value as a valid option to DATM\_MODE in the env\_conf.xml file.

# Appendix B. Using the Script `runinit_ibm.csh` to both Run CLM and Interpolate Datasets

The script `runinit_ibm.csh` can be used on the NCAR bluefire machine to run CLM to create a template file and then run `interpinic` and do this over a variety of standard resolutions. By default it is setup to loop over the following resolutions:

```
foreach res ( "1.9x2.5" "10x15" "4x5" "0.9x1.25" "2.5x3.33" "0.47x0.63" "48x96" )
```

It is also only setup to run CLMCN and only particular masks for each resolution. But, the script can be modified by the user to run over whatever list you would like it to. It is also hooked up to the **build-namelist** XML database, so will only use the datasets that are part of the database, see Chapter 3 to see how to add files to the database. The script runs CLM only using OpenMP threading and as such can be run interactively, but it can also be submitted to the batch que.

# Appendix C. Scripts for testing CLM

Technically, you could use the customization we gave in Chapter 1 to test various configuration and namelist options for CLM. Sometimes, it's also useful to have automated tests though to test that restarts give exactly the same results as without a restart. It's also useful to have automated tests to run over a wide variety of configurations, resolutions, and namelist options. To do that we have several different types of scripts set up to make running comprehensive testing of CLM easy. There are two types of testing scripts for CLM. The first are the CESM test scripts, which utilize the **create\_newcase** scripts that we shown how to use in this User's Guide. The second are a set of stand-alone scripts that use the **CLM configure** and **build-namelist** scripts to build and test the model as well as testing the CLM tools as well. Below we will go into further details of how to use both methods.

## Testing CLM Using the CESM Test Scripts

We first introduce the test scripts that work for all CESM components. We will use the **create\_test** and then the **create\_test\_suite** scripts. The **create\_test** runs a specific type of test, at a given resolution, for a given compset using a given machine. There is a list of different tests, but the "ERI" tests do several things at once, running from startup, as well as doing exact branch and restart tests. So to run "ERI" testing at 2-degree with the I1850CN compset on bluefire you do the following.

```
> cd scripts  
> ./create_test -testname ERI.f19_g16.I1850CN.bluefire  
> cd ERI.f19_g16.I1850CN.bluefire.$id  
> ./ERI.f19_g16.I1850CN.bluefire.$id.build  
> ERI.f19_g16.I1850CN.bluefire.$id.submit
```

When the test is done it will update the file `TestStatus` with either a PASS or FAIL message.

To run a suite of tests from a list of tests with syntax similar to above you use **create\_test\_suite** as follows passing it a ASCII list of tests. There are already some test lists in the `scripts/ccsm_utils/Testlists` directory a few of which are specific to CLM. To run for the CLM bluefire test list, on bluefire, you would do the following:

```
> cd scripts  
> ./create_test_suite -input_list ccsm_utils/Testlists/bluefire.clm.auxtest  
# Submit the suite of tests (note $id refers to the integer job number for this job)  
> ./cs.submit.$id.bluefire  
# Later check the tests with...  
> ./cs.status.$id  
# The above will give a PASS or FAIL message for each test.
```

For more information on doing testing with the CESM scripts see the CESM1.0.4 User's Guide (<http://www.cesm.ucar.edu/models/cesm1.0/cesm>) on testing.

## Testing CLM Using the CLM Stand-Alone Testing Scripts

In the `models/lnd/clm/test/system` directory there is a set of test scripts that is specific to stand-alone CLM. It does testing on configurations harder to test for in the CESM test scripts, and also

allows you to test the CLM tools such as **mkgriddata** and **mksurfdata**. The main driver script is called **test\_driver.sh** and it can run both interactively as well as being submitted to the batch queue. Like other scripts you can get help on it by running the "-help" option as: **test\_driver.sh -help**. There is also a **README** file that gives details about environment variables that can be given to **test\_driver.sh** to change it's operation.

To submit a suite of stand-alone tests to the batch que:

```
> cd models/lnd/clm/test/system
> ./test_driver.sh
```

You can also run tests interactively:

```
> cd models/lnd/clm/test/system
> ./test_driver.sh -i
```

The output of the help option is as follows:

```
ERROR: machine harmon.cgd.ucar.edu not currently supported
```

A table of the list of tests and the machines they are run on is available from: [test\\_table.html](#)  
[\(..../test/system/test\\_table.html\)](#)

## Testing PTCLM use the PTCLM test script

There is a simple test script for PTCLM called **testcases.csh** in the PTCLM directory (**scripts/ccsm\_utils/Tools/lnd/clm/PTCLM**). The test script is setup to run on the machines: bluefire, scd data machines, edinburgh, lynx, yong, and jaguar. You simply run the script interactively. The script will write out the status of tests to a file called: **tc.job#.status**.

There are a few environment variables that can be used with **testcases.csh** to change it's operation.

**CESM\_ROOT**: To test with a separate root to CESM code set this env variable to the root directory to use.  
**CLM\_SOFF**: If set to TRUE -- stop on first failed test rather than continuing to run.  
**CLM\_RETAIN\_FILES**: If set to FALSE -- cleanup tools build first.  
**DEBUG**: If set to TRUE -- setup cases, but do not build or run.

# Appendix D. Building the Users-Guide Documentation for CLM

All of the documentation for CLM can be built using GNU Makefiles that are available in the appropriate directories. The Makefiles require the following utilities: **docbook2html**, **docbook2pdf**, **protex**, and **latex2html**.

To build the Users Guide for CLM (requires docbook).

```
> cd models/lnd/clm/doc/UsersGuide  
> gmake
```

Note, that when the Users-Guide is built it will get output from other CLM utilities that by nature abort, and hence stop the make from continuing. However, this is expected so you should simply run **gmake** again until it either completes or comes upon a legitimate issue. Here is what a sample warning looks like when **gmake** is run.

```
The following line will fail in the make as it calls die -- but that is expected  
Check that the output config_help.tlog is good and redo your make  
..../bld/configure -help >&` config_help.tlog  
make: *** [config_help.tlog] Error 255
```

To build the Code Reference Guide for CLM (requires **protex** and **latex2html**). The make here uses a **Filepath** file that points to the list of directories that you want **protex** to run over. You should examine this file and make sure it is appropriate for what you need to do, before running the make.

```
> cd models/lnd/clm/doc/CodeReference  
> gmake
```

To build the table of tests for the CLM test suite. The make here runs a UNIX shell script to create a html table of the list of tests run on the different machines from the CLM test suite.

```
> cd models/lnd/clm/test/system  
> gmake
```