

Users' Guide to HRLDAS

Version 0.2

Oct 2010

DRAFT

Chapter 1:

Introduction

What is HRLDAS?

The High Resolution Land Data Assimilation System (HRLDAS) runs the Noah Land Surface Model (Noah LSM) in an uncoupled mode (i.e., not coupled with any atmospheric model) to evolve land surface and soil state variables over some time period. The time period for which the Noah LSM is run may be as long as many months or years. From initial conditions of soil temperature, soil moisture, and other state variables, HRLDAS applies the Noah LSM, forced by analyses of atmospheric conditions, shortwave and longwave radiation, and precipitation, to update the land state.

The applications of HRLDAS are often intended to address the issue of the soil state *spinup*, that is, evolving the soil state variables from low-resolution or otherwise uncertain or inadequate initial conditions, through a long integration of Noah LSM forced by relatively well-observed variables, to a state which is well-balanced with respect to the Noah LSM physics, and represents high-resolution soil and vegetation conditions appropriate for the model grid.

The use of Noah LSM in HRLDAS differs from the implementation of Noah LSM coupled within the Weather Research and Forecasting (WRF) model, in which WRF model results of atmospheric conditions, shortwave and longwave radiation, and precipitation are used to update the land state, which in turn influences the WRF simulation of atmospheric conditions. In a coupled implementation (e.g., Noah LSM in WRF), there is a two-way feedback of information: atmospheric computations influence the soil results, and soil computations influence atmospheric results. This is in contrast to the uncoupled implementation (e.g., HRLDAS), in which there is no feedback from the Noah LSM to influence the atmospheric forcing conditions.

HRLDAS is typically run on the horizontal grid of a mesoscale model, but the motivated user

should be able to adapt it for running at individual points or sets of stations. For use with nested model configurations, HRLDAS must be run independently for each nest, and there is no communication of information among nests.

Relationship of HRLDAS to WRF

HRLDAS was designed to be used in conjunction with the WRF model, specifically the Advanced Research WRF (ARW). As such, the Noah LSM source code in HRLDAS is adapted from the code used in WRF. Since certain input files come from the WRF Preprocessing System (WPS) and WRF, running the WRF preprocessors is typically a prerequisite for running HRLDAS. Tasks regarding WRF and its preprocessors will not be treated at length in this document, but details on installing, configuring, and running WRF and its preprocessors may be found at the WRF users web page: <http://www.mmm.ucar.edu/wrf/users>.

Steps for running HRLDAS

There are several steps for running HRLDAS. See Fig. 1 for a graphical representation of these steps.

- 1) Acquire raw data for initialization and forcing
- 2) Extract and organize raw data for initialization and forcing
- 3) Set up the model grid configuration
- 4) Perform spatial and temporal interpolation of initialization and forcing data
- 5) Run the Noah LSM

Steps 4 and 5 are encompassed in HRLDAS. Step 3 is performed through the WPS and WRF programs. Steps 1 and 2 are considered preparation for running HRLDAS, and the tasks involved may vary greatly depending on a user's needs, the available data, and the computing environment. As such, there are no standard programs available which perform steps 1 and 2. However, this document discusses these steps at length, providing some tools and suggestions for achieving steps 1 and 2.

Things to define, consider, or expand upon in this document

- spinup
- Use of file naming templates and replacement strings
- data compression
- wind components
- source model

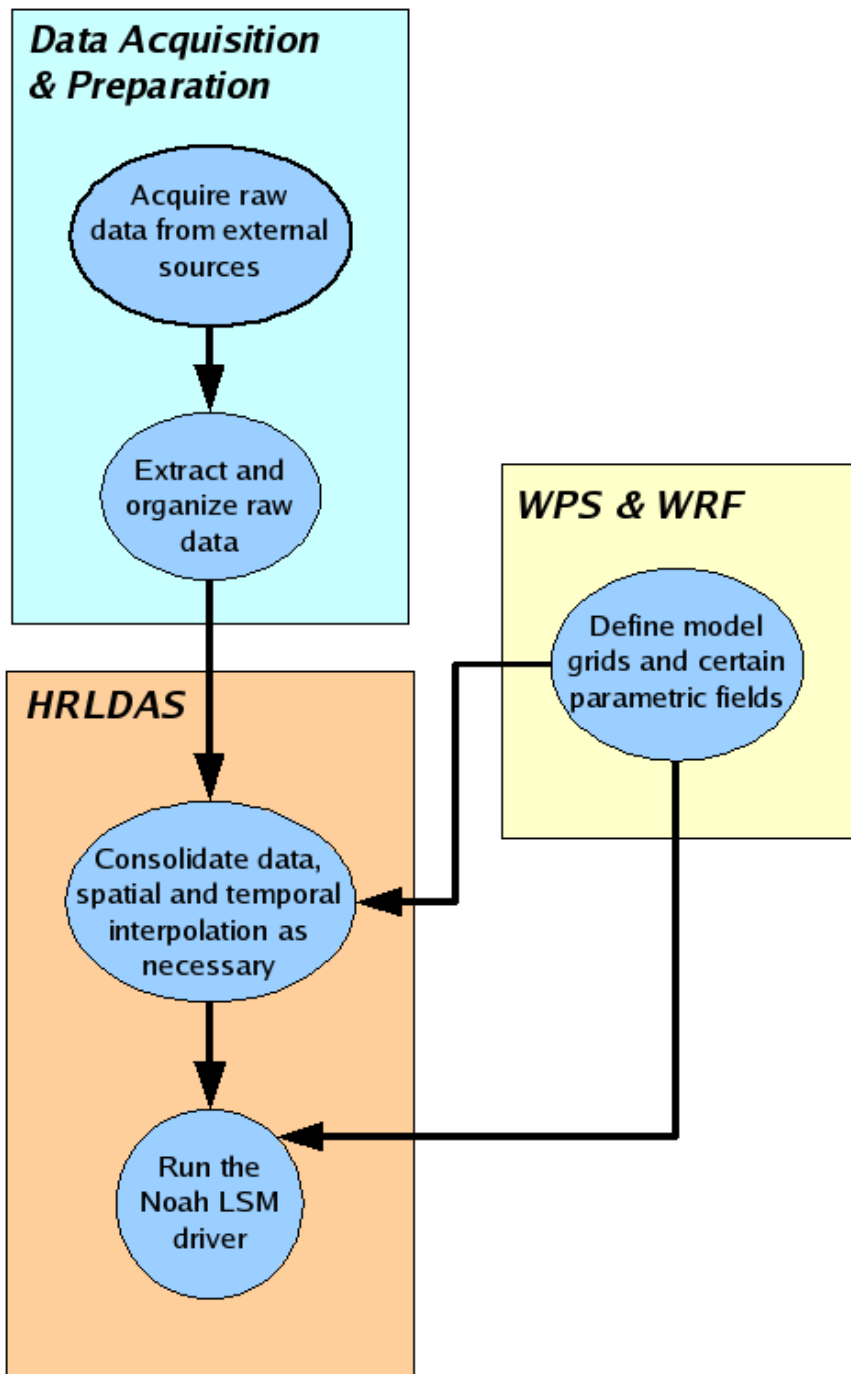


Fig 1: Schematic

- source data
- GRIB (Edition 1)
- GRIB (Edition 2)
- References – especially some journal articles

DRAFT

Chapter 2:

Compiling HRLDAS

This chapter covers the details of compiling the HRLDAS code and associated programs.

What is needed

The HRLDAS has been developed for Unix- or Linux-style environments. The usual functionality of such an environment is necessary (e.g., compilers, linkers, make utility), as well as some specialized libraries and header files (i.e, for the NetCDF and bzip2 packages). Of particular importance are:

Unix/Linux “make” utility

The “make” utility uses recipes spelled out in special files, called makefiles, to build executable software packages from source code. Any full-featured Unix/Linux platform should have a version of “make” available. Unfortunately, some “make” implementations are more equal than others, so in some circumstances, users may find themselves having to modify the makefiles for a successful compilation. For this reason, the makefiles have been kept fairly simple.

Fortran 90/95 compiler

Many different flavors of Fortran are available for Unix/Linux platforms, and they all take different options. Be sure you know what Fortran compiler you are using, and how to find the compiler options it takes (e.g., documentation, man page, etc.).

C compiler

Any full-featured Unix/Linux platform should have a C compiler available.

The C Preprocessor Program “cpp”

Any full-featured Unix/Linux platform should have a version of the C Preprocessor Program available.

NetCDF libraries, header files, and module files for Fortran-90 NetCDF

The primary output from the HRLDAS programs is in NetCDF format. You (or your systems administrator) may have to build the NetCDF package for your platform. Be sure you know where the NetCDF libraries, include files, and modules are installed. The NetCDF package is freely available from <http://www.unidata.ucar.edu/software/netcdf>. Be sure that the NetCDF build is configured to create the Fortran-90 interface. You will generally need to use the same Fortran compiler for building NetCDF as for building the HRLDAS programs (newer NetCDF builds reportedly may not have this restriction).

The (optional) parallel capabilities of HRLDAS require that the NetCDF package be compiled with parallel I/O enabled. This requires the NetCDF4 generation of code, as well as the HDF libraries and the dependencies of HDF. Building and configuring these packages is beyond the scope of this document. Here be dragons.

The Jasper library

The Jasper library is needed for unpacking GRIB Edition 2 data. Even if you only expect to use GRIB Edition 1 data, the source code includes the flexibility for GRIB Edition 2, and expects to have this library available. This library is available from <http://www.ece.uvic.ca/~mdadams/jasper>.

bzip2 libraries and header files (optional)

For certain data files (specifically, the GRIB files used as input to `consolidate_grib.exe`), the programs are able to read the files in a bzip2-compressed form. A full-featured Unix/Linux platform will probably have the bzip2 compression program installed, but may not have the libraries and header files available. You may select whether to use this bzip2 capability when you configure your HRLDAS compilation (see `user_build_options`, below).

Tested platforms

The build procedure and code have been tested (at least minimally) on:

- Linux (CentOS) with Portland Group Inc. Fortran compiler
- Linux (CentOS) with g95 Fortran compiler
- Linux (CentOS) with GNU gfortran Fortran compiler

- Linux (CentOS) with Intel Fortran compiler
- IBM withxlf Fortran compiler

Steps to compile

Untar the HRLDAS tar file

```
tar -xvf HRLDAS-<version>.tar
```

Change to top-level HRLDAS directory

```
cd HRLDAS-<version>
```

Edit the configuration options in file “user_build_options”

See section “user_build_options”, below

Run the make program to build the executables

```
make
```

verify that the build completed

[check existence of executables?]

[make test?]

user_build_options

Many configuration options are specified in the file user_build_options. Chances are good that users will need to modify specific options for their particular systems. Options for several different computer platforms are included in the user_build_options file as examples. Activate an appropriate block of options by un-commenting (removing “#” characters at the beginning of the line).

Configuration and compilation options set through the user_build_options are:

COMPILERF90

The command to invoke the Fortran 90/95 compiler. Consult your system administrator for details.

FREESOURCE

The option for your Fortran compiler to specify free-form source code. This is often something like “-free”, “-freeform”, “-Mfree”. Some compilers may have options for specifying allowable record lengths in free-form source code files. For specifics, consult the documentation for your Fortran compiler.

F90FLAGS

Other Fortran-90 compiler options. This line should include byte-swap I/O options as needed. This is also a handy place to add debugging options, like “-g”, or “-check bounds”, if desired. This is also a good place to experiment with optimization options.

MODFLAG

The Fortran compiler option to specify alternate directories for Fortran module files. This is often “-I” or “-M”. For specifics, consult the documentation for your Fortran compiler.

LD_FLAGS

Additional options to the linker program.

CPP

The command to invoke the C Preprocessor Program. Consult your system administrator for details.

CPPFLAGS

Options passed to the C-Preprocessor Program. HRLDAS makes use of conditional compilation triggered by specific CPPFLAGS options:

- D_HRLDAS_URBAN_: Compiles code for using the Single-Layer Urban Canopy Model with HRLDAS.
- D_PARALLEL_: Compiles code for multi-processor execution of HRLDAS.

LIBS

Additional libraries needed.

LIBJASPER

The Fortran compiler options that specify the pathname and the library filename for the Jasper libraries (used for GRIB Edition 2 compression/decompression) and the JPEG library (used by the Jasper library). The actual locations of these libraries depends on how these libraries were installed. Consult your system administrator for details.

E.g.: `LIBJASPER = -L/usr/lib -ljpeg -L/usr/local/jasper/lib \`
`-ljasper`

INCJASPER

The compiler options that specify the pathname to the directory that contains the Jasper include files. As with the Jasper libraries, the actual locations of these include files depends on how the Jasper library was installed. Consult your system administrator for details.

E.g.: INCJASPER = -I/usr/local/jasper/include

NETCDFMOD

The Fortran compiler option that specifies the pathname to the directory that contains the NetCDF Fortran-90 module files. Consult your systems administrator for details.

E.g.: NETCDFMOD = -I/usr/local/netcdf/include

NETCDFLIB

The Fortran compiler options that specify the path to the directory that contains the NetCDF libraries, and the specific library to use. Consult your systems administrator for details.

E.g.: NETCDFLIB = -L/usr/local/netcdf/lib -lnetcdf

HDF5LIB

The Fortran compiler options that specify the path to the directory that contains the HDF5 libraries, and specific libraries to load. Consult your systems administrator for details. This option is required only if you want to use the parallel capabilities of HRLDAS.

E.g.: HDF5LIB = -L/usr/local/HDF5/lib -lhdf5_hl -lhdf5 -lz

BZIP2

Enable the use of bzip2 libraries to read bzip2-compressed files.

E.g.: BZIP2 = YES or BZIP2 = NO

BZIP2_LIB

Flags to load the bzip2 libraries.

E.g.: BZIP2_LIB = -L/usr/lib -lbz2

BZIP2_INCLUDE

Flags to direct the compiler to the BZIP2 header files.

E.g.: BZIP2_INCLUDE = -I/usr/include

RM

The command to remove files.

E.g.: RM = rm -f

CC

The command to invoke the C compiler.

Chapter 3:

Required data

Initialization fields

A variety of fields are needed for initialization and forcing of HRLDAS. For initialization, the required fields are:

- Soil temperature (at multiple levels)
- Soil moisture (at multiple levels)
- Canopy water content
- Skin temperature
- Water equivalent of accumulated snow depth

The initialization fields are generally needed only at the initial time. See chapter ?? for further details regarding the initialization fields.

Forcing fields

For forcing, the required fields are:

- Near-surface air temperature
- Source-model terrain elevation (for adjusting near-surface air temperature to destination-model surface elevation).
- Near-surface atmospheric mixing ratio
- Near-surface u-component of the horizontal wind (grid-relative)

- Near-surface v-component of the horizontal wind (grid-relative)
- Surface pressure
- Precipitation rate
- Downward shortwave radiation flux at the surface
- Downward longwave radiation flux at the surface

Forcing fields are required at each time step throughout the HRLDAS assimilation period. See chapter ?? for further details regarding the forcing fields.

Parametric fields [is this an appropriate term?]

Other fields define soil, vegetation, or other geographical parameters, and may be referred to as parametric fields. These fields typically come from the files produced by the WRF preprocessing. These fields are:

- Latitude
- Longitude
- Vegetation category
- Soil category
- Green vegetation fraction
- A minimum green vegetation fraction for a typical year at a given point
- A maximum green vegetation fraction for a typical year at a given point
- A time-invariant deep-soil temperature

Parametric fields are needed at the initial time, and certain fields may be updated during HRLDAS integration as appropriate. See chapter ?? for further details regarding parametric fields.

Chapter 4:

Acquire and organize data

Acquiring and organizing the data for HRLDAS is probably the most difficult and time-consuming task in running HRLDAS. The data typically come from operational model analyses (and sometimes model forecasts), and if not collected in real time, must be retrieved from storage in long-term archives. The organization of the data in these long-term archives, and indeed in the operational files collected in real time, is not necessarily suitable for use in HRLDAS, so significant reorganization of data is likely necessary before it can be used in HRLDAS.

After much experimentation, it was decided that the most appropriate way to organize the raw data is to store each 2-d field at each time as a single GRIB file containing a single GRIB (Edition 1 or Edition 2) record. Because GRIB is a WMO standard for archiving and distributing gridded meteorological data, many of the fields required by HRLDAS are already available in GRIB format, and throughout the community many tools are available for manipulating and visualizing GRIB datasets. Additionally, as by design the GRIB format enables significant data compression, GRIB datasets can use significantly less disk space than many other formats, an important consideration when many months or years of data are required.

Since GRIB datasets are not usually distributed or archived as single records, some manipulation of and extraction from the GRIB datasets is generally required of the user. A simple GRIB extraction program *gribextract* is provided for this purpose. Some knowledge of GRIB is required for effective use of this program (See the not-yet-written appendix: *gribextract*).

Full date and time information must also be determined from the GRIB file name, either in the full path name of the file or the name of the individual file itself.

Aside from those two stipulations (i.e., that each 2-d field reside in its own GRIB file, that the date and time information be available in the file or path name), there are few other conventions for naming

or organizing files. Users may choose to organize their data in a way best suited to their own applications or their own personal tastes. Some thought to organizing data is advisable, because for a long term HRLDAS job, hundreds of thousands of GRIB files may be necessary.

Examples:

```
/disk1/user1/data/temperature/2007011518.grb  
/disk1/user1/storage/2007011518/temperature.grb  
/disk1/user1/hrldas/data/2007/01/15/18/T.2007011518.grb  
/disk1/user1/rawdata/2007/01/15/2m_temp.18Z.grb  
/disk1/user1/HRLDAS/GRIB/2007011500/T.2007011518.grb  
/disk1/user1/2007011400/TMP.2007011518.grb  
/disk1/user1/data/2007/01/15/T.2007011518.grb
```

GRIBEXTRACT

Program `gribextract` is an experimental utility for extracting single GRIB records from GRIB-formatted files. This program (or a tool with similar functionality) will be essential in organizing data for use with HRLDAS. This program is available in the `HRLDAS/Utility_programs` subdirectory, and may be compiled by executing “`make`” in that directory. Proper use of this program requires familiarity with the data and familiarity with the GRIB format specification. Care must be taken with this program to insure that only single fields are extracted, as per the HRLDAS convention that each 2D field reside in its own GRIB file. For instructions on using this program, invoke “`gribextract -help`”.

Chapter 5:

Prepare model configuration for HRLDAS

HRLDAS is based around the implementation of Noah LSM for the Weather Research and Forecasting Model (WRF). As such, HRLDAS requires certain files and fields as prepared for WRF. The easiest way to create these files for HRLDAS is to prepare data for input into the WRF model, i.e., run the WRF Preprocessing System (WPS) and WRF program *real.exe*.

The strategy is to set up model grids and model input files as for a WRF simulation, but then use these files to run HRLDAS instead of WRF.

HRLDAS does not directly handle nesting, but HRLDAS can be run independently for multiple domains. If nested HRLDAS data are desired, WPS and the WRF program *real.exe* must be run for multiple domains.

GEO_EM Files

The WPS program *geogrid.exe* will create files named *geo_em.d<nn>*, where *<nn>* is a two-digit grid identification number. For a single domain execution, *geogrid.exe* will create file *geo_em.d01*. For a three domain execution, *geogrid.exe* will create files *geo_em.d01*, *geo_em.d02*, and *geo_em.d03*. These are NetCDF-formatted files, using the NetCDF conventions of WRF input and output files [i.e., not any widely recognized NetCDF convention, like COADS]. For details on running WPS, see [WRF documentation link].

Only a few pieces of information from the *geo_em* files are needed for HRLDAS. A surrogate *geo_em* file, containing only these minimal data, will work fine for the purposes of HRLDAS. The data needed from the *geo_em* files are:

Required NetCDF dimensions in the geo_em file:

west_east

The number of grid cells in the X direction.

south_north

The number of grid cells in the Y direction.

month

The number of months in the year (12), used as a dimension for the monthly mean green vegetation fraction variable, which has one field per month.

Time

The number of time periods in the NetCDF file; should be equal to 1. The Time dimension is the unlimited dimension in the NetCDF dataset.

Required NetCDF global attributes in the geo_em file:

MAP_PROJ

An integer flag identifying the map projection used. Three map projections are recognized:

MAP_PROJ=1 :: Lambert Conformal

MAP_PROJ=2 :: Polar Stereographic

MAP_PROJ=3 :: Mercator

DX

The grid-cell size (m) in the X direction.

DY

The grid-cell size (m) in the Y direction. DY is usually equal to DX.

TRUELAT1

A “true” latitude, or standard parallel, used in the definition of some map projections, which indicates where the plane of the projection intersects the earth. TRUELAT1 is used for the Polar Stereographic and Lambert Conformal map projections.

TRUELAT2

A second “true” latitude, or standard parallel, used in the definition of the Lambert Conformal map projection. For the Lambert Conformal map projection, TRUELAT2 indicates a second latitude (in addition to TRUELAT1) where the plane of the projection intersects the earth.

STAND_LON

The central longitude of the map projection (not necessarily of the map grid itself) used in the definition of some map projections. *STAND_LON* is the longitude that is parallel to the Y-axis.

MMINLU

A string identifying the source of the land-use categories, which determines the mapping from integer land-use categories to physical land-use characteristics. The only formally recognized value for *MMINLU* is “USGS”.

Required NetCDF fields in the geo_em file:

For the variables in the NetCDF files, no variable attributes necessary. However, attributes such as “description” and “units” strings are highly recommended.

XLAT_M (or XLAT)

2-dimensional latitude field (dimensioned west_east by south_north).

XLONG_M (or XLONG)

2-dimensional longitude field (dimensioned west_east by south_north).

GREENFRAC

Monthly climatological green vegetation fraction (dimensioned west_east by south_north by 12 months).

As an example, sample output from using the NetCDF `ncdump` utility to describe the details of a minimal *geo_em* file is included:

Example: Minimal geo_em contents

```
netcdf geo_em {
dimensions:
    Time = UNLIMITED ; // (1 currently)
    month = 12 ;
    south_north = 239 ;
    west_east = 279 ;
variables:
    float GREENFRAC(Time, month, south_north, west_east) ;
    float XLAT_M(Time, south_north, west_east) ;
    float XLONG_M(Time, south_north, west_east) ;

// global attributes:
    :DX = 12000.f ;
    :DY = 12000.f ;
    :TRUELAT1 = 30.f ;
    :TRUELAT2 = 60.f ;
    :STAND_LON = -98.f ;
    :MAP_PROJ = 1 ;
    :MMINLU = "USGS" ;
}
```

WRFINPUT Files

The WRF program *real.exe* prepares files for input to the WRF model. The WPS programs (after *geogrid.exe*) and the WRF program *real.exe* require time-dependent fields to set up a forecast for a specific date. However, since HRLDAS uses only constant fields from these files, the actual time run for WPS and *real.exe* is immaterial. Simply select a convenient set of meteorological data to run through the WRF preprocessing programs.

Similarly to WPS, *real.exe* may be run for multiple domains, creating files called, e.g., *wrfinput_d01*, *wrfinput_d02*, *wrfinput_d03*. HRLDAS uses a certain subset data from these *wrfinput* files; again, similarly constructed surrogate files containing this minimal amount of data will work well for HRLDAS.

Required NetCDF Dimensions in the wrfinput file:

west_east

The number of grid cells in the X direction. This must be the same as the *west_east* dimension in the *geo_em* file.

south_north

The number of grid cells in the Y direction. This must be the same as the west_east dimension in the geo_em file.

Required NetCDF global attributes in the wrfinput file:

GRID_ID or *grid_id*

Integer flag identifying the grid.

MAP_PROJ

An integer flag identifying the map projection used. This must be the same as the MAP_PROJ attribute in the geo_em file. Three map projections are recognized:

MAP_PROJ=1 :: Lambert Conformal

MAP_PROJ=2 :: Polar Stereographic

MAP_PROJ=3 :: Mercator

STAND_LON

The central longitude of the map projection (not necessarily of the map grid itself) used in the definition of some map projections. STAND_LON is the longitude that is parallel to the Y-axis. This attribute must have the same value as the STAND_LON attribute in the geo_em file

TRUELAT1

A “true” latitude, or standard parallel, used in the definition of some map projections, which indicates where the plane of the projection intersects the earth. TRUELAT1 is used for the Polar Stereographic and Lambert Conformal map projections. This attribute must have the same value as the TRUELAT1 global attribute in the geo_em file.

TRUELAT2

A second “true” latitude, or standard parallel, used in the definition of the Lambert Conformal map projection. For the Lambert Conformal map projection, TRUELAT2 indicates a second latitude (in addition to TRUELAT1) where the plane of the projection intersects the earth. This attribute must have the same value as the TRUELAT2 global attribute in the geo_em file.

DX

The grid-cell size (m) in the X direction. This attribute must have the same value as the DX global attribute in the geo_em file.

DY

The grid-cell size (m) in the Y direction. *DY* is usually equal to *DX*. This attribute must have the same value as the *DY* global attribute in the *geo_em* file.

ISWATER

The soil category which corresponds to water points.

ISURBAN

The vegetation category which corresponds to urban points.

Required NetCDF Fields in the WRFINPUT file:

XLAT (or *XLAT_M*)

2-dimensional latitude field (degrees North). *XLAT* is dimensioned (west_east by south_north). This field must be the same as the *XLAT_M* field in the *geo_em* file.

XLONG (or *XLONG_M*)

2-dimensional longitude field (degrees East). *XLONG* is dimensioned west_east by south_north. This field must be the same as the *XLONG_M* field in the *geo_em* file.

HGT

2-dimensional surface elevation field (m). *HGT* is dimensioned west_east by south_north.

IVGTYP

2-dimensional array of vegetation category indices. *IVGTYP* is dimensioned west_east by south_north.

ISLTYP

2-dimensional array of soil category indices. *ISLTYP* is dimensioned west_east by south_north.

TMN

2-dimensional constant deep-soil temperature field (K) *TMN* is dimensioned west_east by south_north)

As an example, sample *ncdump* output of a minimal *wrfinput* file is included:

Example: Minimal wrfinput contents:

```
netcdf wrfinput {
dimensions:
    Time = UNLIMITED ; // (1 currently)
    south_north = 239 ;
    west_east = 279 ;
variables:
    float HGT(Time, south_north, west_east) ;
    int ISLTYP(Time, south_north, west_east) ;
    int IVGTYP(Time, south_north, west_east) ;
    float TMN(Time, south_north, west_east) ;
    float XLAT(Time, south_north, west_east) ;
    float XLONG(Time, south_north, west_east) ;

// global attributes:
    :DX = 12000.f ;
    :DY = 12000.f ;
    :GRID_ID = 1 ;
    :TRUELAT1 = 30.f ;
    :TRUELAT2 = 60.f ;
    :STAND_LON = -98.f ;
    :MAP_PROJ = 1 ;
    :MMINLU = "USGS" ;
    :ISWATER = 16 ;
    :ISURBAN = 1 ;
}
```

DR

Chapter 6:

Data Consolidation Step

Program consolidate_grib.exe

This chapter describes the steps needed to perform the data consolidation step, i.e., running program `consolidate_grib.exe`. The discussion in this chapter assumes that the user has prepared appropriate model-grid-related files (`geo_em` and `wrfinput`), that data have been retrieved from original sources, and that the GRIB files for these data have been organized in an appropriate manner as described [above]. This chapter also assumes that successful compilation of the appropriate programs has been performed according to [compiling chapter].

The data consolidation step is performed by program `consolidate_grib.exe`. This program accesses the myriad source data files for various initialization, forcing, and parametric fields, performs necessary temporal and horizontal interpolation, and writes the data out into “LDASIN” files, one per hour.

Input to `consolidate_grib.exe`

- Namelist file `namelist.input`
- `wrfinput_d###` or similarly constructed file, as identified in the `namelist.input` file
- `geo_em.d###` or similarly constructed file, as identified in the `namelist.input` file
- GRIB-formatted source data as identified in the `namelist.input` file
- Tables mapping GRIB code numbers to field names, units and descriptions, available in directory `HRLDAS/HRLDAS_COLLECT_DATA/GRIB_TABLES`. The full path to this directory must be set as environment variable `GRIB_ROOT`. [I hope to be able to remove this

necessity in future releases. It's kind of messy.]

Environment Variable

An environment variable, “GRIB_ROOT”, must be set to the full path to the directory containing GRIB tables. [I hope to be able to remove this necessity in future releases. It's kind of messy.] E.g.:
`setenv GRIB_ROOT /d1/user1/HRLDAS/HRLDAS_COLLECT_DATA/GRIB_TABLES.`

Options in *namelist.input* file

Program `consolidate_grib.exe` is controlled primarily through the use of a namelist file. The name of a namelist file may be given as a command-line argument to `consolidate_grib.exe`. If no command-line argument is given, `consolidate_grib.exe` will look for a file called *namelist.input*. The various options to set in the *namelist.input* file are:

STARTDATE

The starting date (format <YYYYMMDDHH>) of the period for data consolidation.

ENDDATE

The ending time (format <YYYYMMDDHH>) of the period for data consolidation.

DataDir

A top-level directory name from which other directory names may be built. This is simply a convenience variable for the filename templates below, where the string “<DataDir>” in the filename templates below may be replaced by the value set here for *DataDir*.

Rainfall_interp

Default: 0

An integer flag specifying the procedure used to remap the rainfall fields to the model grid. (Default 1). Two options are available, 0 and 1:

RAINFALL_INTERP=0: Take the nearest-neighbor value.

RAINFALL_INTERP=1: Subdivide the source grid cell into some number of smaller cells, and allocate precipitation from the source grid to the destination grid based on accumulating values from the sub-cells which fall into the destination grid cell.

RAINFALL_INTERP option 0 may be appropriate, and may save much processing time, when the HRLDAS grid is somewhat smaller than the source grid. RAINFALL_INTERP option 1 is not appropriate when the HRLDAS grid is somewhat larger than the source grid.

Full_ic_freq

Default: 0

Integer specifying the frequency (in hours) for full initial conditions for HRLDAS to be written to the HRLDAS input files. Value of 0 indicates that full initial conditions are to be prepared only for the time indicated by option STARTDATE. Preparing full initial conditions more often than that consumes extra processing power and disk space, but does allow for greater flexibility in choosing the starting time of the Noah LSM integration. Set this to a positive value (preferably some integer multiple of 24) if you need the flexibility of starting the Noah LSM integration at various times. Set this to -1 to turn off the generation of extra fields needed for initial conditions.

Rescale_shortwave

Default: .FALSEE.

Logical specifying whether the program will attempt to correct a half-hour time offset that has been noted in the GCIP shortwave radiation analyses. Set this to false if the time offset does not exist, or if you are using a different source of shortwave radiation analyses.

Geo_em_flnm

The full path name of the geo_em file created by the geogrid program of the WRF Preprocessing System (WPS) package. This file is needed to supply certain geographic information and parametric fields which may not be available in the wrfinput file.

Wrfinput_flnm

The name of the wrfinput file created as in preparation for running the WRF model. This file is needed to supply certain geographic information and parametric fields which may not be available in the geo_em file.

Zfile_template

A file name template for building the full path name of the GRIB file which contains the model elevation field of the model or analysis package that supplies the temperature field. This file is needed to perform a height-based temperature adjustment from the temperature source model terrain elevation to the HRLDAS terrain elevation.

Tfile_template

The file name template for building the full path name of the GRIB file which contains the near-surface (generally 2-m) air temperature field.

Ufile_template

The file name template for building the full path name of the GRIB file which contains the near-surface (generally 10-m) u-component (grid relative) of the horizontal wind field.

Vfile_template

The file name template for building the full path name of the GRIB file which contains the near-surface (generally 10-m) v-component (grid relative) of the horizontal wind field.

Pfile_template

The file name template for building the full path name of the GRIB file which contains the surface pressure field [or is this near-surface, e.g., 2-m?] [Should this be adjusted for terrain elevation like the 2-m T data?].

Qfile_template

The file name template for building the full path name of the GRIB file which contains the near-surface (generally 2-m) atmospheric mixing ratio field.

LWfile_template

The file name template for building the full path name of the GRIB file which contains the surface downward longwave radiation flux field.

WEASDfile_template

The file name template for building the full path name of the GRIB file which contains the water equivalent of the accumulated snow depth field.

CANWTfile_template

The file name template for building the full path name of the GRIB file which contains the canopy water field.

LANDSfile_template

The file name template for building the full path name of the GRIB file which contains the land/sea mask from model of the original source data for the soil fields.

SKINTfile_template

The file name template for building the full path name of the GRIB file which contains the skin temperature field

STfile_template

The file name template for building the full path name of the GRIB file which contains the soil temperature field.

SMfile_template

The file name template for building the full path name of the GRIB file which contains the soil moisture field.

SWfile_primary

The file name template for building the full path name of the GRIB file which contains the preferred surface downward shortwave radiation flux field.

SWfile_secondary

The file name template for building the full path name of the GRIB file which contains a secondary surface downward shortwave radiation flux field, to be used for filling in holes when the primary shortwave radiation dataset has missing values.

PCPfile_primary

The file name template for building the full path name of the GRIB file which contains the preferred precipitation rate field.

PCPfile_secondary

The file name template for building the full path name of the GRIB file which contains a secondary precipitation rate field, to be used for filling in holes when the primary precipitation dataset has missing values.

Vtable

The namelist file also contains a non-namelist section, a table of mappings from GRIB (Edition 1 and Edition 2) code numbers to field name and unit strings. This table is similar to the “Vtables” used in WPS. The beginning of this section is marked by the string “<VTABLE>”, and the end of this section is marked by the string “</VTABLE>”.

The first four columns of the Vtable describe the GRIB Edition 1 code numbers for referencing a variable. The last four columns describe the GRIB Edition 2 code numbers for referencing a variable. With these two sets of columns, the program is adaptable to read both GRIB Edition 1 and GRIB Edition 2, and also to read GRIB datasets which may not use standard code numbers.

The “NAME” column defines the name by which the variable is recognized internally by the *con-*

solidate_grib.exe program. These names (as in the example below) should not be changed, as the program does not recognize other strings.

File name templates

This program makes use of file name templates to build actual file names. Special strings in the file name templates enclosed between “<” and “>” are replaced with appropriate data to build the final file names. This approach allows for great flexibility for the user in organizing the multitudinous data files. The strings recognized for replacement are:

<*DataDir*>

The string “<*DataDir*>” found in a file name template will be replaced by the value specified in the *DataDir* namelist option.

<*YYYY*>

The string “<*YYYY*>” found in a file name template will be replaced by the four-digit year of the time currently being processed.

<*MM*>

The string “<*MM*>” found in a file name template will be replaced by the two-digit month of the time currently being processed.

<*DD*>

The string “<*DD*>” found in a file name template will be replaced by the two-digit day of the time currently being processed.

<*HH*>

The string “<*HH*>” found in a file name template will be replaced by the two-digit hour of the time currently being processed.

<*date*>

The string “<*date*>” found in a file name template will be replaced by the ten-digit date of the time currently being processed (format <YYYYMMDDHH>).

<*init+nn*>

The string “<*init+nn*>”, where nn is a two digit number (either 00, 12, 24, or 36), will be replaced by the ten-digit forecast initialization time (format <YYYYMMDDHH>) which would have a forecast hour between nn and nn+12 [or maybe nn-12 and nn?] valid at the time currently being processed.

A sample namelist is provided as an example here: An sample namelist is provided as an example here:

```

&files
STARTDATE      = "2007-12-12_00"
ENDDATE        = "2008-12-14_00"
DataDir        = "/wig/kmanning/hrldas/data"

geo_em_flnm    = "/dl/kmanning/hrldas/grids/geo_em.d01.nc"
wrfinput_flnm  = "/dl/kmanning/hrldas/grids/wrfinput_d01"

rainfall_interp = 0
full_ic_frq    = 0

Zfile_template = "<DataDir>/NAM/<init+12>/SURFACE_ELEVATION.<date>.grb"
               "<DataDir>/NAM/<init-12>/SURFACE_ELEVATION.<date>.grb"
               "<DataDir>/NAM/<init-24>/SURFACE_ELEVATION.<date>.grb"
               "<DataDir>/NAM/<init-36>/SURFACE_ELEVATION.<date>.grb"

Tfile_template = "<DataDir>/NAM/<init+12>/T.<date>.grb",
               "<DataDir>/NAM/<init-12>/T.<date>.grb",
               "<DataDir>/NAM/<init-24>/T.<date>.grb",
               "<DataDir>/NAM/<init-36>/T.<date>.grb",

Ufile_template = "<DataDir>/NAM/<init+12>/U.<date>.grb",
               "<DataDir>/NAM/<init-12>/U.<date>.grb",
               "<DataDir>/NAM/<init-24>/U.<date>.grb",
               "<DataDir>/NAM/<init-36>/U.<date>.grb",

Vfile_template = "<DataDir>/NAM/<init+12>/V.<date>.grb",
               "<DataDir>/NAM/<init-12>/V.<date>.grb",
               "<DataDir>/NAM/<init-24>/V.<date>.grb",
               "<DataDir>/NAM/<init-36>/V.<date>.grb",

Pfile_template = "<DataDir>/NAM/<init+12>/P.<date>.grb",
               "<DataDir>/NAM/<init-12>/P.<date>.grb",
               "<DataDir>/NAM/<init-24>/P.<date>.grb",
               "<DataDir>/NAM/<init-36>/P.<date>.grb",

Qfile_template = "<DataDir>/NAM/<init+12>/Q.<date>.grb",
               "<DataDir>/NAM/<init-12>/Q.<date>.grb",
               "<DataDir>/NAM/<init-24>/Q.<date>.grb",
               "<DataDir>/NAM/<init-36>/Q.<date>.grb",

LWfile_template = "<DataDir>/NAM/<init+12>/LW.<date>.grb",
               "<DataDir>/NAM/<init-12>/LW.<date>.grb",
               "<DataDir>/NAM/<init-24>/LW.<date>.grb",
               "<DataDir>/NAM/<init-36>/LW.<date>.grb",

WEASDfile_template = "<DataDir>/NAM/<init+12>/WEASD.<date>.grb",
                   "<DataDir>/NAM/<init-12>/WEASD.<date>.grb",
                   "<DataDir>/NAM/<init-24>/WEASD.<date>.grb",
                   "<DataDir>/NAM/<init-36>/WEASD.<date>.grb",

CANWTfile_template = "<DataDir>/NAM/<init+12>/CANWAT.<date>.grb",
                   "<DataDir>/NAM/<init-12>/CANWAT.<date>.grb",
                   "<DataDir>/NAM/<init-24>/CANWAT.<date>.grb",

LANDSfile_template = "<DataDir>/NAM/<init+12>/LANDSEA.<date>.grb",
                   "<DataDir>/NAM/<init-12>/LANDSEA.<date>.grb",
                   "<DataDir>/NAM/<init-24>/LANDSEA.<date>.grb",

SKINTfile_template = "<DataDir>/NAM/<init+12>/SKINTEMP.<date>.grb",
                   "<DataDir>/NAM/<init-12>/SKINTEMP.<date>.grb",
                   "<DataDir>/NAM/<init-24>/SKINTEMP.<date>.grb",

STfile_template = "<DataDir>/NAM/<init+12>/SOIL_T_000-010.<date>.grb",
                 "<DataDir>/NAM/<init+12>/SOIL_T_010-040.<date>.grb",
                 "<DataDir>/NAM/<init+12>/SOIL_T_040-100.<date>.grb",
                 "<DataDir>/NAM/<init+12>/SOIL_T_100-200.<date>.grb",
                 "<DataDir>/NAM/<init-12>/SOIL_T_000-010.<date>.grb",
                 "<DataDir>/NAM/<init-12>/SOIL_T_010-040.<date>.grb",
                 "<DataDir>/NAM/<init-12>/SOIL_T_040-100.<date>.grb",
                 "<DataDir>/NAM/<init-12>/SOIL_T_100-200.<date>.grb",

SMfile_template = "<DataDir>/NAM/<init+12>/SOIL_M_000-010.<date>.grb",
                 "<DataDir>/NAM/<init+12>/SOIL_M_010-040.<date>.grb",
                 "<DataDir>/NAM/<init+12>/SOIL_M_040-100.<date>.grb",
                 "<DataDir>/NAM/<init+12>/SOIL_M_100-200.<date>.grb",

```

```

"<DataDir>/NAM/<init-12>/SOIL_M_000-010.<date>.grb",
"<DataDir>/NAM/<init-12>/SOIL_M_010-040.<date>.grb",
"<DataDir>/NAM/<init-12>/SOIL_M_040-100.<date>.grb",
"<DataDir>/NAM/<init-12>/SOIL_M_100-200.<date>.grb",

SWfile_primary = "<DataDir>/SRB/SW.<date>30.grb",

SWfile_secondary = "<DataDir>/NAM/<init+12>/SW.<date>.grb",
"<DataDir>/NAM/<init-12>/SW.<date>.grb",
"<DataDir>/NAM/<init-24>/SW.<date>.grb",

PCPfile_primary = "<DataDir>/ST4/<YYYY><MM><DD>/ST4.<date>.grb",

PCPfile_secondary = "<DataDir>/NAM/<init-12>/PCP.<date>.grb",
"<DataDir>/NAM/<init-24>/PCP.<date>.grb",
"<DataDir>/NAM/<init-36>/PCP.<date>.grb",
/

```

<VTABLE>

GRIB1 Param	Level Type	From Level1	To Level2	Name	Units	Description	GRIB2 Discp	GRIB2 Catgy	GRIB2 Param	GRIB2 Level
11	105	2		T2D	K	Temperature at 2 m	0	0	0	103
51	105	2		Q2D	kg kg{-1}	Specific Humidity at 2 m	0	1	0	103
33	105	10		U2D	m s-1	U at 10 m	0	2	2	103
34	105	10		V2D	m s-1	V at 10 m	0	2	3	103
1	1	0		PSFC	Pa	Surface Pressure	0	3	0	1
144	112	0	10	SMOIS_1	kg m-3	Soil Moist 0-10 cm below grn layer (Up)	2	0	192	106
144	112	10	40	SMOIS_2	kg m-3	Soil Moist 10-40 cm below grn layer	2	0	192	106
144	112	40	100	SMOIS_3	kg m-3	Soil Moist 40-100 cm below grn layer	2	0	192	106
144	112	100	200	SMOIS_4	kg m-3	Soil Moist 100-200 cm below gr layer	2	0	192	106
85	112	0	10	STEMP_1	K	T 0-10 cm below ground layer (Upper)	2	0	2	106
85	112	10	40	STEMP_2	K	T 10-40 cm below ground layer (Upper)	2	0	2	106
85	112	40	100	STEMP_3	K	T 40-100 cm below ground layer (Upper)	2	0	2	106
85	112	100	200	STEMP_4	K	T 100-200 cm below ground layer (Bottom)	2	0	2	106
91	1	0		SEAICE	proprtn	Ice flag	10	2	0	1
81	1	0		LANDSEA	proprtn	Land/Sea flag (1=land,0=sea in NAM)	2	0	0	1
11	1	0		SKINTEMP	K	Skin temperature (can use for SST also)	0	0	0	1
61	1	0		RAINRATE	kg m-2	Accumulated precipitation	0	1	8	1
65	1	0		WEASD	kg m-2	Water equivalent snow depth	0	1	13	1
223	1	0		CANWAT	kg m-2	Plant Canopy Surface Water	2	0	196	1
204	1	0		SWDOWN	W m-2	Downward short-wave radiation flux	0	4	192	1
205	1	0		LWDOWN	W m-2	Downward long-wave radiation flux	0	5	192	1
7	1	0		TERRAIN	m	Source model terrain elevation	0	3	5	1

</VTABLE>



Chapter 7:

Forcing Fields

This chapter describes the forcing fields used in HRLDAS. For the final HRLDAS step, the fields must be available for each hour of the simulation period. The original data sources may provide data at intervals greater than one hour. For use in HRLDAS, the `consolidate_grib.exe` program can interpolate to the required hourly time interval.

Air Temperature

Over the Continental United States, the NCEP North American Model (NAM) or NAM Data Assimilation System (NDAS) may provide the appropriate data records at 3-hourly intervals.

Atmospheric Mixing Ratio

Over the Continental United States, the NCEP North American Model (NAM) or NAM Data Assimilation System (NDAS) may provide the appropriate data records at 3-hourly intervals.

U-Component of the Horizontal Wind

Over the Continental United States, the NCEP North American Model (NAM) or the NAM Data assimilation System (NDAS) may provide the appropriate data records at 3-hourly intervals.

V-Component of the Horizontal Wind

Over the Continental United States, the NCEP North American Model (NAM) or the NAM Data assimilation System (NDAS) may provide the appropriate data records at 3-hourly intervals.

Surface Pressure

Over the Continental United States, the NCEP North American Model (NAM) or NAM Data Assimilation System (NDAS) may provide the appropriate data records at 3-hourly intervals.

Precipitation Rate

Over the Continental United States, the NCEP Stage IV precipitation analysis may provide the appropriate data at hourly intervals.

Shortwave Radiation

Over the Continental United States, the GEWEX Continental Scale International Project (GCIP) and GEWEX Americas Prediction Project (GAPP) Surface Radiation Budget (SRB) Data provides hourly fields of downward shortwave radiation flux at the surface. This dataset is available from <http://www.atmos.umd.edu/~srb/gcip>. [Do I want to get into the SW questions here?]

Longwave Radiation

Over the Continental United States, the NCEP NAM provides 6-hourly (forecast) fields of longwave radiation flux at the surface.

Chapter 8:

Initialization Fields

This chapter discusses the fields used to initialize HRLDAS.

Soil Temperature

The soil temperature in four layers. The [currently-supported] layers are from 0.0 to 0.1 m below ground, 0.1 to 0.4 m, 0.4 to 1.0 m, and 1.0 to 2.0 m. Over the Continental United States, the North American Model (NAM) or the NAM Data Assimilation System (NDAS) may provide appropriate fields.

Soil Moisture

The volumetric soil moisture, in the same layers as the soil temperature. Over the Continental United States, the North American Model (NAM) or the NAM Data Assimilation System (NDAS) may provide appropriate fields.

Canopy Water

The water stored in the plant canopy. Over the Continental United States, the North American Model (NAM) or the NAM Data Assimilation System (NDAS) may provide appropriate fields.

Skin Temperature

Over the Continental United States, the North American Model (NAM) or the NAM Data Assimilation System (NDAS) may provide appropriate fields.

Water Equivalent of the Snow Depth

Over the Continental United States, the North American Model (NAM) or the NAM Data Assimilation System (NDAS) may provide appropriate fields.

DRAFT

Chapter 9: Parametric Fields

This chapter describes some of the fields best described as surface, soil, or vegetation parameters.

VEGFRA

GVFMIN

GVFMAX

Z2D

TMN (TBOT)

Chapter 10:

HRLDAS Executable Noah_hrldas_beta

This chapter details running the HRLDAS executable Noah_hrldas_beta. This chapter assumes that all data collection and preparation steps have been performed, that the HRLDAS program has been properly compiled, that the user has available a full suite of LDASIN files, and that the user has available the corresponding wrfinput file (or similarly constructed file).

Input to Noah_hrldas_beta

Input to the Noah_hrldas_beta program are:

namelist.hrldas

<YYYYMMDDHH>.LDASIN_DOMAIN<domain>

Various *.TBL files

wrfinput_d0<domain> or similarly-constructed file

(optional) RESTART.<YYYYMMDDHH>_DOMAIN<domain>

Running Noah_hrldas_beta

The basic steps to running the HRLDAS executable are to configure the appropriate namelist options in file “namelist.hrldas”, and run the executable “Noah_hrldas_beta”. Both of these files are available in directory HRLDAS/Run. Change to that directory, edit the namelist.hrldas file (as described below in section X.X), and run the Noah_hrldas_beta executable. A few notes on specific ca-

pabilities and options:

Restart capability

The restart capability is a useful feature for HRLDAS, particularly since HRLDAS can run for a long time and create a lot of output. The restart capability saves a snapshot of the HRLDAS model state at regular intervals, allowing a later HRLDAS execution to pick up from a restart time as if rerunning from the original start time. One use for this might be in the case of HRLDAS for a real-time system, in which the restart capability may be used daily to extend prior HRLDAS jobs as new data become available.

Output frequency controls

Because HRLDAS can create a lot of output, it is often desirable to tune the output frequency for specific needs. For example, an HRLDAS spinup may run for several months or even years, during which time little to no history output would be used. One might run the spinup with history output turned off completely (or perhaps restricted to once every few days or weeks), and restart files written once every few weeks. Then when the period of true interest is reached, a restart may be configured to produce history output every hour (or every few hours).

Subwindowing

A subwindowing capability has been built in, to allow HRLDAS to run on a smaller subdomain rather than the full grid. This is useful for testing and debugging situations, where running on a few points (or even a single point) can be convenient.

Urban Canopy Model

The Urban Canopy Model takes a separate namelist record in the same namelist.hrlldas file. The Urban Canopy Model is activated (or deactivated) by the SF_URBAN_PHYSICS flag in the namelist file. The program must be configured at compile time (via user_build_options option CPPFLAGS) to compile the urban code.

Parallel capability

HRLDAS may be run in a parallel configuration, using MPICH. The program must be compiled with MPICH, using the parallel I/O capabilities of NetCDF4 and HDF5.

HRLDAS namelist file “namelist.hrlldas”

The namelist file “namelist.hrlldas” is the primary user control over execution of the HRLDAS executable, Noah_hrlldas_beta. The following namelist variables in file “namelist.hrlldas” are user options

to control various aspects of job execution.

HRLDAS_CONSTANTS_FILE

The full path name to the wrfinput file (or a similarly-constructed file). This is a required variable; there is no valid default value. For details about the format and content of this file, see Appendix 1, HRLDAS_CONSTANTS_FILE)

INDIR

The full path name to the directory that contains the LDASIN files for a particular HRLDAS run. This is a required variable; there is no valid default value.

OUTDIR

The full path name to the directory to which the LDASOUT output files will be written for a particular HRLDAS run. This is an optional variable; the default is to write data to the directory from which the program is invoked.

START_YEAR

The 4-digit year of the start time of the HRLDAS run. This is a required variable; there is no valid default value.

START_MONTH

The 2-digit month of the start time of the HRLDAS run. This is a required variable; there is no valid default value.

START_DAY

The 2-digit day of the start time of the HRLDAS run. This is a required variable; there is no valid default value.

START_HOUR

The 2-digit hour of the start time of the HRLDAS run. This is a required variable; there is no valid default value.

EXTERNAL_FPAR_FILENAME_TEMPLATE

A template used to build filenames which may provide FPAR maps to the HRLDAS program. This is optional. Don't worry about this for now. For experimental use – special circumstances only.

EXTERNAL_LAI_FILENAME_TEMPLATE

A template used to build filenames which may provide LAI maps to the HRLDAS program. This

is optional. Don't worry about this for now. For experimental use – special circumstances only.

RESTART_FILENAME_REQUESTED

The full pathname of the restart file to use when restarting the HRLDAS integration. This is an optional variable; the default is a blank string indicating that this is not a restart run. If the restart file is specified by the exact name, it must exist for the program to run. If the restart file includes the string “<LATEST>”, the program will attempt to find the latest restart file (by the date specified in the file name) that matches the requested string, with “<LATEST>” being replaced with the 10-character date string in the form YYYYMMDDHH.

E.g.: RESTART_FILENAME_REQUESTED= /user1/io/RESTART_2010041300_DOMAIN1

E.g.: RESTART_FILENAME_REQUESTED= /user1/io/RESTART_<LATEST>_DOMAIN1

NOAH_TIMESTEP

The time step (seconds) to use for the Noah LSM integration. If this is different from the FORCING_TIMESTEP, temporal interpolation will be performed between the available LDASIN files.

FORCING_TIMESTEP

The time increment (seconds) between the available forcing files created by program consolidate_grib.

OUTPUT_TIMESTEP

The time increment (seconds) at which history output will be created. This must be an integer multiple of the NOAH_TIMESTEP set above.

KHOUR

Number of hours to run HRLDAS, counting from the start time (or the restart time in case of a restart run). This is an optional variable; but there is no valid default value, and exactly one of KHOUR and KDAY must be set. If the input data available in directory <INDIR> runs out before KHOUR hours have passed, the Noah_hrldas_beta executable will stop with a worrisome failure message, but this is to be expected.

KDAY

Number of days to run HRLDAS, counting from the start time (or the restart time in case of a restart run). This is an optional variable; but there is no valid default value, and exactly one of KHOUR and KDAY must be set. If the input data available in directory <INDIR> runs out before KDAY days have passed, the Noah_hrldas_beta executable will stop with a worrisome failure

message, but this is to be expected.

RESTART_FREQUENCY_HOURS

How frequently (in hours) to create an HRLDAS restart file. The value, converted to seconds, must be an integer multiple of the NOAH_TIMESTEP. This is an optional variable; the default value is 0 (zero) indicating that no restart files are to be created.

HISTORY_OUTPUT

How frequently to create HRLDAS history output (LDASOUT) files. This is an optional variable; the default value is 0 (zero) indicating that no history output files are to be created.

SPLIT_OUTPUT_COUNT

How many output time periods to write in a single history output (LDASOUT) file, before starting a new output file. This is an optional variable; the default is 1.

SUBWINDOW_XSTART

Starting x-coordinate of a subwindow to use instead of the full grid. This is an optional variable; the default is to run the full domain (no subwindowing).

SUBWINDOW_XEND

Ending x-coordinate of a subwindow to use instead of the full grid. This is an optional variable; the default is to run the full domain (no subwindowing).

SUBWINDOW_YSTART

Starting y-coordinate of a subwindow to use instead of the full grid. This is an optional variable; the default is to run the full domain (no subwindowing).

SUBWINDOW_YEND

Ending y-coordinate of a subwindow to use instead of the full grid. This is an optional variable; the default is to run the full domain (no subwindowing).

IZOTLND

Controls the setting of the CZIL term. This is an optional variable; the default value is 0.

SFCDIF_OPTION

Option to use newer (SFCDIF_OPTION=1) or older (SFCDIF_OPTION=0) SFCDIF routine. This is an optional variable; the default value is 0.

UPDATE_SNOW_FROM_FORCING

Option to activate or deactivate updating the snow-cover fields from available analyses. This is an optional variable; the default value is .TRUE.

SF_URBAN_PHYSICS

Integer flag to use (1) or skip (0) the Single Layer Urban Canopy Model. This is an optional variable; the default value is 0, to skip the Urban Canopy Model.

ZLVL_URBAN

Height (m) of atmospheric fields for Urban Canopy Model. This is an optional variable; but required if the Urban Canopy Model is used (i.e., if UCMCALL is set to 1).

DRAFT