## Contents

1. **Introduction** 1  
   1.1 This Document 1  
   1.2 Acknowledgments 1  

2. **NU-WRF System** 3  
   2.1 Components 3  
   2.2 Other Features 6  
   2.3 History 8  

3. **Obtaining Software** 23  
   3.1 Software Usage Agreement 23  
   3.2 Tar File 24  
   3.3 Subversion Repository 24  
   3.4 Directory Structure 25  

4. **Building Software** 26  
   4.1 Compilers 26  
   4.2 External Libraries and Tools 26  
   4.3 Build System 27  

5. **Front-End Workflows** 31  
   5.1 Basic Workflow 31  
   5.2 Land Surface Initialization and LIS Coupling 32  
   5.3 Use of GEOS-5 Meteorological Data 35  
      5.3.1 GEOS2WRF 36  
      5.3.2 MERRA2WRF 47  
   5.4 Use of New Erodible Soil Options 51  
   5.5 Use of GOCART Aerosol Data 53  
   5.6 Use of CASA CO$_2$ Data 59  
   5.7 Use of SPoRT Sea Surface Temperature Data 61  
   5.8 Use of RSS Sea Surface Temperature Data 61  
   5.9 Initializing Small Inland Lake Temperatures 63  

6. **Post-Processors** 65  
   6.1 G-SDSU 65  
   6.2 RIP4 66  
   6.3 ARWPOST 66  
   6.4 UPP 66  
   6.5 MET 67  
   6.6 LVT 67  

A. **Frequently Asked Questions** 69  

B. **Porting NU-WRF** 75
CONTENTS

References 80
1 Introduction

1.1 This Document

This is the NASA Unified-Weather Research and Forecasting (NU-WRF) Version 8 User’s Guide. This document provides an overview of NU-WRF; describes how to download, compile, and run the NU-WRF software; and provides guidance on porting the software to new platforms.

This document consists of six sections and two appendices:

- Section 1 is the present introductory section.
- Section 2 provides general information about the NU-WRF project, components, and development history.
- Section 3 provides information on obtaining a software usage agreement and the NU-WRF source code.
- Section 4 describes how to compile the NU-WRF software;
- Section 5 describes several front-end workflows that can be employed with the NU-WRF modeling system, ranging from basic weather simulation to advanced aerosol-microphysics-radiation coupling to alternate initialization methods. This includes information on new pre-processors and NASA changes to the community WRF model.
- Section 6 describes several post-processors for visualization and/or verification.
- Finally, Appendix A answers Frequently Asked Questions about NU-WRF, while Appendix B provides guidance on porting NU-WRF to new platforms.

1.2 Acknowledgments

The development of NU-WRF has been funded by the NASA Modeling, Analysis, and Prediction Program. The Goddard microphysics, radiation, aerosol coupling modules, and G-SDSU are developed and maintained by the Mesoscale Atmospheric Processes Laboratory at NASA Goddard Space Flight Center (GSFC). The GOCART and NASA dust aerosol emission modules are developed and maintained by the GSFC Atmospheric Chemistry and Dynamics Laboratory. The LIS, LDT, and LVT components are developed and maintained by the GSFC Hydrological Sciences Laboratory. The CASA2WRF, GEOS2WRF, GOCART2WRF, LISCONFIG, MERRA2WRF, NDVIBARENESS4WRF, and SST2WRF components are maintained by the GSFC Computational and Information Sciences and Technology Office. SST2WRF includes binary reader source code developed by Remote Sensing Systems.

Past and present contributors affiliated with the NU-WRF effort include: Kristi Arsenault, Clay Blankenship, Scott Braun, Rob Burns, Jon Case, Mian

The upstream community WRF, WPS, and ARWPOST components are developed and supported by the National Center for Atmospheric Research (NCAR), which is operated by the University Corporation for Atmospheric Research and sponsored by the National Science Foundation (NSF). The Kinetic Pre-Processor included with WRF-Chem was primarily developed by Dr Adrian Sandu of the Virginia Polytechnic Institute and State University. The community RIP4 is maintained by NCAR and was developed primarily by Dr Mark Stoelinga, formerly of the University of Washington. The UPP is developed by the National Centers for Environmental Prediction; the community version is maintained by the Developmental Testbed Center (DTC), which is sponsored by the National Oceanic and Atmospheric Administration, the United States Air Force, and the NSF. DTC also develops and maintains the community MET package. The community PREP CHEM SOURCES is primarily developed by the Centro de Previsao de Tempo e Estudos Climaticos, part of the Instituto Nacional de Pesquisas Espaciais, Brazil.

NU-WRF Version 8 “Bjerknes” is named after Dr Vilhelm Bjerknes (developer of the Bjerknes Circulation Theorem and founder of the groundbreaking Bergen School of Meteorology) and his son Dr Jacob Bjerknes (a developer of the classic Norwegian Cyclone Model).
2 NU-WRF System

NU-WRF has been developed at Goddard Space Flight Center (GSFC) as an observation-driven integrated modeling system representing aerosol, cloud, precipitation, and land processes at satellite-resolved scales [Peters-Lidard et al. (2015)]. NU-WRF is intended as a superset of the standard NCAR Advanced Research WRF [WRF-ARW; Skamarock et al. (2008)] and incorporates:

- The GSFC Land Information System [LIS; see Kumar et al. (2006) and Peters-Lidard et al. (2007)], coupled to WRF and also available as a stand-alone executable;
- The WRF-Chem enabled version of the Goddard Chemistry Aerosols Radiation Transport model [GOCART; Chin et al. (2002)];
- GSFC radiation and microphysics schemes including revised couplings to the aerosols [Shi et al. (2014); Lang et al. (2014)]; and
- The Goddard Satellite Data Simulator Unit [G-SDSU; see Matsui et al. (2014)].

In addition, multiple pre- and post-processors from the community and from GSFC have been collected with WRF and LIS. Taken together, the NU-WRF modeling system provides a framework for simulating aerosol and land use effects on such phenomena as floods, tropical cyclones, mesoscale convective systems, droughts, and monsoons (Peters-Lidard et al., 2015). Support also exists for treating CO$_2$ as a tracer, with plans to further refine into source components (anthropogenic versus biogenic). Finally, the software has been modified to use netCDF4 (http://www.unidata.ucar.edu/software/netcdf/) with HDF5 compression (https://www.hdfgroup.org/HDF5/), reducing netCDF file sizes by up to 50%.

Work is underway to incorporate NU-WRF in a Maximum Likelihood Ensemble Filter-based atmospheric data assimilation system, with the capability of assimilating cloud and precipitation affected radiances (Zhang et al., 2015). In addition, some secondary, rarely used elements of the community WRF modeling system that are not yet included with NU-WRF will be added in the future.

2.1 Components

The NU-WRF package contains the following components:

- The WRFV3/ component contains a modified copy of the core WRF version 3.7.1 modeling system [see Chapter 5 of NCAR (2016)], the WRF-Fire wildfire library [see Appendix A of NCAR (2016)], the WRF-Chem atmospheric chemistry library [see Peckham et al. (2015a) and Peckham et al. (2015b)], and several preprocessors (REAL, CONVERT_EMISS, TC, and
2.1 Components

NDOWN). These codes have been modified to add the 2014 Goddard radiation package [Matsui and Jacob, 2014], the new Goddard 4ICE microphysics scheme [Lang et al., 2014], the new Goddard 3ICE microphysics scheme [Shi et al., 2014], couplings between these schemes and GOCART, new dust emission options, a new CO$_2$ tracer parameterization, and several new diagnostic products. The legacy community WRF versions of the Goddard microphysics and radiation schemes are separate options from the latest versions developed by GSFC. WRFV3/ also includes LIS 7.1rp7 (in WRFV3/lis/) with code modifications to both LIS and WRF-ARW to facilitate on-line coupling between the atmosphere and Noah land surface models, as well as land data assimilation (NASA, 2015).

- The **WPS/ component contains a modified copy of the WRF Preprocessing System version 3.7.1** [see Chapter 3 of NCAR (2016)]. This includes the GEOGRID, UNGRIB, and METGRID programs used to set up a WRF domain, to interpolate static and climatological terrestrial data, to extract fields from GRIB and GRIB2 files, and to horizontally interpolate the fields to the WRF grid. This also includes the optional AVG_TSFCC preprocessor, which calculates time-average surface temperatures for use in initializing small in-land lake temperatures.

- The **ldt/ component contains version 7.1rp2 of the NASA Land Data Toolkit (LDT) software** (NASA, 2015a). This acts both as a preprocessor for LIS (including interpolation of terrestrial data to the LIS grid and separate preprocessing for data assimilation) and a postprocessor for LIS (merging dynamic fields from a LIS off-line “spin-up” simulation with static data for eventual input to WRF in LIS coupled mode).

- The **utils/lisconfig/ component contains the NASA LISCONFIG software** for customizing LDT and LIS ASCII input files so their domain(s) (grid size, resolution, and map projection) match that of WRF. It uses output from the WPS GEOGRID program to determine the reference latitude and longitude.

- The **utils/geos2wrf/ component contains version 2 of the NASA GEOS2WRF software**, which extracts and/or derives atmospheric data from the Goddard Earth Observing System Model Version 5 [Rienecker et al., 2008; http://gmao.gsfc.nasa.gov/GMAO_products] for input into WRF. It also contains MERRA2WRF, which can preprocess atmospheric fields from the Modern-Era Retrospective Analysis for Research and Applications [MERRA; Rienecker et al., 2011] hosted by the Goddard Earth Sciences Data Information Services Center (GES DISC; http://daac.gsfc.nasa.gov), as well as MERRA-2 reanalyses [Bosilovich et al., 2015] also hosted by GES DISC. These programs essentially take the place of UNGRIB in WPS, as UNGRIB cannot read the netCDF, HDF4, or HDFEOS formats used with GEOS-5, MERRA, and MERRA-2.
2.1 Components

- The `utils/sst2wrf/` component contains the NASA SST2WRF preprocessor, which reads sea surface temperature (SST) analyses produced by Remote Sensing Systems (http://www.remss.com) and converts into a format readable by the WPS program METGRID. This essentially takes the place of UNGRIB as the SST data are in a non-GRIB binary format.

- The `utils/ndviBareness4Wrf/` component contains the NASA NDVIBARENESS4WRF preprocessor, which reads gridded Normalized Difference Vegetation Index (NDVI) data and calculates a “surface bareness” field for use by the NU-WRF dynamic dust emissions scheme. Currently the software reads NDVI produced from the NASA Global Inventory Modeling and Mapping Studies (GIMMS) group and by the NASA Short-term Prediction Research and Transition Center (SPoRT). Both products are based on observations from the Moderate Resolution Imaging Spectroradiometer (MODIS), which is flown on the NASA Terra and Aqua satellites. The preprocessor outputs both the NDVI and the derived bareness fields in a format readable by the WPS program METGRID.

- The `utils/prep_chem_sources/` component contains a modified copy version of the community PREP_CHEM_SOURCES version 1.5 preprocessor. This program prepares anthropogenic, biogenic, wildfire, and volcanic emissions data for further preprocessing by the WRF-Chem preprocessor CONVERT_EMISS. The NU-WRF version of PREP_CHEM_SOURCES uses the WPS map projection software to ensure consistency in interpolation. It also adds support for GFEDv3.1 biomass burning emissions [see van der Werf et al. (2010) and Randerson et al. (2013)], NASA QFED wildfire emissions [Darmenov and da Silva 2013], support for new 72-level GOCART background fields, improved interpolation of the GOCART background fields when the WRF grid is at a relatively finer resolution, and output of data for plotting with the NASA PLOT_CHEM program.

- The `utils/plot_chem/` component stores a simple NCAR Graphics based PLOT_CHEM program for visualizing the output from PREP_CHEM_SOURCES. This program is only intended for manual review and sanity checking, not for publication quality plots.

- The `utils/gocart2wrf_2/` component stores version 2 of GOCART2WRF, a NASA program for reading GOCART aerosol data from an offline GOCART run (Chin et al. 2002), online GOCART with GEOS-5, MERRAero (Kishcha et al. 2014), or MERRA-2 (Bosilovich et al. 2015) files; interpolate the data to the WRF grid; and then add the data to netCDF4 initial condition and lateral boundary condition (IC/LBC) files for WRF. This includes a script for downloading and processing MERRA-2 files from the GES DISC web page.

- The `utils/casa2wrf/` component contains the NASA CASA2WRF preprocessor and related software to read CO$_2$ emissions and concentrations from
the CASA biosphere model, and interpolate and append the data to the WRF netCDF4 IC/LBC files [Tao et al. 2014].

- The RIP4/ component contains a modified copy of the NCAR Graphics-based Read/Interpolate/Plot (Stoelinga 2006) graphical postprocessing software, version 4.6.7. Modifications include support for UMD land use data (from LIS) and WRF-Chem output.

- The ARWpost/ component contains version 3.1 of the GrADS-compatible ARWpost program for visualization of output [see Chapter 9 of NCAR (2016)].

- The UPP/ component contains a modified copy of the NCEP Unified Post Processor version 3.0. This software can derive fields from WRF netCDF output and write in GRIB format (see DTC (2015b)).

- The MET/ component contains version 5.1 of the Meteorological Evaluation Tools (DTC, 2015a) software produced by the Developmental Testbed Center. This can be used to evaluate WRF atmospheric fields converted to GRIB via UPP against observations and gridded analyses.

- The lvt/ component contains version 7.1rp1 of the NASA Land Verification Toolkit (NASA, 2015c) for verifying LIS land and near-surface fields against observations and gridded analyses.

- The GSDSU/ component contains version 3.5.1 of the Goddard Satellite Data Simulation Unit (Matsui and Kemp 2016), which can be used to simulate satellite imagery, radar, and lidar data for comparison against actual remote-sensing observations.

In addition, NU-WRF includes a unified build system written in the Bash scripting language to ease compilation of the different NU-WRF components, and to automatically resolve several dependencies between the components (e.g., WPS requires WRFV3 to be compiled first). The build system is discussed in section 4.3.

2.2 Other Features

While NU-WRF aims to be a superset of the official WRF modeling system, there are some WRF components that are currently not supported. These are:

- WRF-Hydro, a hydrologic library for WRF (see https://www.ral.ucar.edu/projects/wrf_hydro). The source code for WRF-Hydro is located in WRFV3/hydro/.

- WRFDA, the data assimilation library for WRF [see Barker et al. (2012), Chapter 6 of NCAR (2016), and http://www2.mmm.ucar.edu/wrf/users/wrfd/]. The source code for WRFDA is not included in NU-WRF Bjerknes.
• OBSGRID, an objective analysis program for WRF [see Chapter 7 of NCAR (2016)]. The source code for this program is not included with NU-WRF.

• PROC_OML, a preprocessor for interpolating HYCOM [see Bleck et al. (2002) and https://hycom.org] 3D ocean model temperature data, and for writing for further processing by the WPS METGRID program. See Chapter 10 of NCAR (2016). The source code for this program is not included with NU-WRF.

• WRF-NMM, an alternative dynamic core developed by NOAA/NCEP. The NMM source code is located in WRFV3/dyn_nmm/ but there is no support to compile it with the NU-WRF build system. While it may be possible to run WRF-NMM with the physics packages and coupling added by NASA, it has never been tested and is not recommended. (Note that as of May 2016, WRF-NMM is no longer supported by the Developmental Testbed Center.)

• HWRF, the Hurricane WRF modeling system developed by NOAA/NCEP with on-line coupling between WRF-NMM and the Princeton Ocean Model for Tropical Cyclones (see http://www.dtcenter.org/HurrWRF/users/index.php). This is a standalone software package provided by the Developmental Testbed Center, and is not included in NU-WRF.

• READ_WRF_NC, a utility similar to the netCDF NCDUMP program for examining WRF netCDF output [see Chapter 10 of NCAR (2016)]. The source code is not included in NU-WRF.

• IOWRF, a utility for manipulating WRF-ARW netCDF data [see Chapter 10 of NCAR (2016)]. The source code is not included in NU-WRF.

• WRF_INTERP, a new utility to interpolate WRF data to isobaric, constant height (AGL or MSL), isentropic, or constant equivalent potential temperature levels. The source code is not included in NU-WRF.

• P_INTERP, a utility to interpolate WRF netCDF output to user-specified isobaric levels [see Chapter 10 of NCAR (2016)]. The source code is not included in NU-WRF.

• V_INTERP, a utility to add vertical levels to a WRF-ARW netCDF file [see Chapter 10 of NCAR (2016)]. The source code is not included in NU-WRF.

In addition, the NU-WRF build system (discussed in Section 4.3) does not currently support compilation of any “ideal” data case [described in Chapter 4 of NCAR (2016)], nor compilation with OpenMP or hybrid OpenMP-MPI. The latter decision is due to a lack of OpenMP support in some of the physics packages.
2.3 History

- Version 8 Patch 1 (v8p1-wrf371-lis71rp7; “Bjerknes Patch 1”)
  
  - **Critical patch.** Changed REAL to always copy TSK and TMN from the grid data structure tsk.gc and tmn.gc members before merging LIS data. This corrects a fatal error when processing METGRID and LIS data without a separate sea surface temperature field.
  
  - **Critical patch.** Added TLAG variable to WRF restart file (via Registry.EM.COMMON). This patch was released by NCAR on 8 Sep 2016 to fix a problem with dynamic deep soil temperature in WRF 3.7, 3.7.1, 3.8, and 3.8.1.
  
  - Updated external libraries:
    
    * New libraries. Added GDAL 2.1.1 and FORTRANGIS 2.5. These are used by the new NDVIBARENESS4WRF utility.
    
    * Upgraded to ESMF 6.3.0rp1, FREETYPE 2.6.5, G2CLIB 1.6.0, GRIB_API 1.17.0, GSL 2.2.1, HDF4 4.2.12, HDF5 1.8.17, NETCDF 4.4.1, NETCDF-Fortran 4.4.4, and PIXMAN 0.34.0.
    
    * NUWRFLIB library package on Discover and Pleiades upgraded to version 8r2.
  
  - UPP patches:
    
    * Now reads REFL_10CM variable from WRF netCDF output when processing radar reflectivity from 3ICE or 4ICE microphysics scheme, instead of searching for non-existent DBZ variable.
    
    * Removed redundant deallocation statements for lightning product, which were causing UPP to crash.
  
  - Added NDVIBARENESS4WRF utility, which can process GIMMS or SPoRT MODIS NDVI products and create a “bareness” field for dynamic dust emissions.
  
  - Added fetch_sport_sst_northwestHemi.py script to download real-time SPoRT SST GRIB2 files. Script and configuration file are in the new scripts/fetch_data directory.
  
  - Updated UNGRIB to set all negative SST values to -1.E30 (the default missing flag used by METGRID). This allows processing the SPoRT SST product without changing the default missing flag.
  
  - LISCONFIG patches:
    
    * Map projection now output by Fortran program LISWRFDOMAIN for insertion into ldt.config file.
    
    * lisWrfDomain.py now checks to see if any domain settings returned by the Fortran program are unused when updating ldt.config and lis.config. This is assumed to indicate an error with the input
2.3 History

file templates (i.e., a configuration setting is missing in the template), which can easily happen if the template is for a different map projection when what WRF will use.

– Added initialization of dx and dy variables in METGRID when processing cylindrical equidistant (lat-lon) or Gaussian data. This fixes an interpolation problem.
– Changed WPS INT2NC utility to output data in netCDF4.
– Minor bug fixes to sample batch configuration files.

• Version 8 (v8-wrf371-lis71rp7; Official “Bjerknes” Release)

– Merged WRF 3.7.1, WPS 3.7.1, PREP CHEM SOURCES 1.5, RIP 4.6.7, UPP 3, and MET 5.1, plus community patches released through May 2016.
– Merged LIS 7.1rp7 and LDT 7.1rp2, including fixes for Mercator map projection.
– Upgraded Intel compilers to 15.0.3.187. Default MPI implementation on both Discover and Pleiades is SGI MPT. Intel MPI 5.1.2.150 is also supported on Discover (for legacy Sandy Bridge nodes). GFORTRAN 4.9.0 and OpenMPI 1.7.3 are temporarily supported for debugging (specifically with VALGRIND). Retired Intel MPI 4 support on Pleiades.
– Removed -xSSE4.2 hardware specific Intel compiler flag. Tests show no consistent performance improvement.
– Modified build system to require MPI for WRF, WPS, and UPP (no serial builds permitted).
– External libraries are now packaged in separate NUWRFLIB collection. This NU-WRF release links against NUWRFLIB 8r1 on Discover and Pleiades. This includes new dependencies (CAIRO, FREETYPE, GHOSTSCRIPT, and PIXMAN) and updates to older dependencies.
– Added sample namelist, lis.config, and ldt.config files for multiple test cases. The LIS cases include use of SRTM 30 sec topography and “slope-aspect” correction which improves the spin-up of soil moisture.
– Updated SST2WRF processing: Changed output mask to separately indicate “low confidence” and “high confidence” SST points. “Low confidence” SST data are now included in output (originally were flagged as bad, but this has a high false alarm rate in the C3VP test case).
– Updated METGRID processing: SST is now interpolated with parabolic, bi-linear, weighted 4-pt average, or weighted 16-pt average (tried in order until one works). When processing TAVGSFC (from AVG_TSFC
2.3 History

preprocessor), METGRID tries parabolic before bi-linear interpolation. This helps reduce data voids and discontinuities in (blended) SST field.

- Updated REAL processing:
  * Preserves LIS greenness and albedo data but processes time-varying LAI data from GEOGRID and METGRID.
  * Changed SST processing to check for obviously bad ocean SSTs when 30s MODIS+Lakes land data are used. Also, disabled use of TAVGSFC as a replacement for bad ocean SST (it seems better to just use skin temperature in this case).

- Added option to skip compiling of WRF CLM4 code. CLM4 is not used by GSFC, and this saves about 10 minutes of build time.

- WRFV3 files module_first_rk_step_part1.F and module_comm_dm_4.f90 are now on the “do not optimize” list to cut down on build times (saves about 10 minutes).

- Multiple bug fixes to prevent memory corruption with arrays allocated with size 1 instead of with WRF “memory” dimensions.

- Updated GEOGRID.TBL files to make GEOGRID.TBL.ARW_NUWRF a strict superset of GEOGRID.TBL.ARW, with the latter now a strict superset of GEOGRID.TBL.ARW.

- Updated dynamic dust emissions. The temperature of the second soil layer is now used instead of skin temperature to detect frozen soil and shut off emissions (this fixes a diurnal cycle bias). Also, the threshold for snow depth is raised to 0.01 m (was zero).

- Increased geopotential height threshold to discriminate between oceans and lakes in sea salt emission calculation.

- Connected the GOCART sulfate chemistry with the RADM2 gas phase chemistry to account for the heterogeneous conversion of $\text{SO}_2$ to sulfate. Previously such conversion was not sufficiently represented in RADM2.

- Redistribution of BC and OC emissions into BC1, BC2, OC1, OC2 aerosol species. This way the NU-WRF aerosol algorithm will be in line with the global GOCART version.

- Added handling of both cumulative diabatic heating (K) and “snapshot” diabatic heating rates (K/s) from 3ICE and 4ICE microphysics schemes. Cumulative heating is now written to restart file.

- 4ICE microphysics updates:
  * Change to “melting” diabatic heating rate to account for wet hail accretion.
  * Adjusted intercept scaling factors are now used to calculate reflectivity.
2.3 History

- 3ICE microphysics update: Bug fix for warm rain code.
- Upgraded GSDSU to version 3.5.1. Changes include:
  * Add support for Thompson microphysics scheme.
  * Visible-IR simulator is more stable (less run-time crashes).
  * New statistical model for CRM diagnosis.
  * Updated GSDSU User Guide.
- CASA2WRF updated to add bug fixes for the following:
  * CO$_2$ flux output files now has CO$_2$ flux for time $t$ and flux tendency between time $t$ and time $t+1$.
  * Ocean CO$_2$ and fossil fuel data is on the corner of the grid instead of the middle point of the grid. All other components are at the midpoint of the grid.
- Added symbolic link to GODDARDRAD_SSLUT in WRFV3/run directory.
- Added fetch_rss_sst.sh script to download RSS SST data (more fault tolerant than Run_SST.csh),
- Added batch scripts to run WPS utility program AVG_TSFC.
- Updated Discover SLURM batch scripts to remove reference to sp1 constraint (no longer exists).
- Multiple bug fixes to initialize variables.

- Version 7-3.5.1-p7 (Official “Arthur” Patch Release 7)
  - Upgrades to LIS 7.1rp5, LDT 7.1rp1, and LVT 7.1rp1. Key changes include:
    * Updates to MERRA2 and MERRA-Land met forcing readers to work with certain LIS time step lengths. Also updates start and end dates for some MERRA2 streams.
    * Corrections to GDAS T170 domain and terrain data.
    * Updates to NLDAS-2 met forcing reader (improves rainfall and convective rainfall around island/coastal areas when downscaled to higher resolutions.)
    * Fix to pcp met forcing field when also reading snow.
    * Updated user guides.
  - Added taucldc and taucldi to Goddard and CAM radiation package listings in Registry.EM_COMMON. These arrays are expected by the radiation schemes to be allocated with the WRF “memory” bounds, but WRF was allocating them with shape (1,1,1) instead. This caused memory corruptions.
  - NUWRF 3ICE and 4ICE microphysics now output reflectivity as REFL_10CM.
2.3 History

– Removed DBZ array from grid data structure to save space. Updated sample RIP template sfcDBZUV.in to use REFL\_10CM instead.
– COMDBZ now calculated in microphysics driver for every microphysics scheme where REFL\_10CM is available.
– Updated Run\_MERRA2.csh to use new path for MERRA2 data on Discover.
– Updated proc\_merra2\_ges\_disc.sh to use new filename for MERRA2 terrain file on GES DISC web page.
– Retired old build configuration files for SLES 11 SP1 nodes on Discover (these no longer exist). Also removed obsolete DISCOVER\_INSTRUCTIONS.TXT.

• Version 7-3.5.1-p6 (Official “Arthur” Patch Release 6)
  – WRF now passes pure LWDOWN (downward longwave radiation) to LIS without first multiplying by surface emissivity.
  – Changed LIS Noah 2.7.1 to multiply LWDOWN by surface emissivity (already done in newer Noah versions).
  – LIS exported surface emissivity now copied back to WRF array for input to radiation.
  – Added capability for CASA\_WRF to temporally interpolate the CO\(_2\) flux depending on the NU-WRF state. CASA\_WRF can now:
    * Read component CO\(_2\) fluxes and interpolate to WRF grid.
    * Read WRF output.
    * Interpolate the CO\(_2\) flux components based on shortwave radiation and surface temperature from WRF state, compute the net production of CO\(_2\), and compute the total flux and tendency.
    * Create input flux files from WRF-Chem runs with CASA CO\(_2\).

• Version 7-3.5.1-p5 (Official “Arthur” Patch Release 5)
  – Updated 4ICE microphysics scheme, and added aerosol coupling.
  – Revised wrfout variable list to reduce file size and write times. Also added total latent heat rate calculation, and now passes dx to 4ICE scheme.
  – Added MPI-safe convective-stratiform index diagnostic.
  – Fixed asynchronous I/O (“quilting”) for coupled WRF-LIS.
  – Changed REAL to use less memory when reading from LDT/LIS netCDF file.
  – LIS updates:
    * Combined lisconfig\_offline, lisconfig\_coupled1, and lisconfig\_coupled2 routines into lisconfig\_generic.
2.3 History

- Added VIIRS and SPoRT GVF testcases.
- Added support to diagnose greenness in units of percentage.
- Updated ldt.config files for the wrfout testcase.
- Added nest loop to metForcGen_interp_finalize routine.
- GOCART2WRF updated to support MERRA-2 GOCART and offline GOCART data.
- Added `proc_merra2_gocart_ges_disc.sh` script to download MERRA-2 GOCART data from NASA GES DISC web page and process with GOCART2WRF.
- Added `proc_merra2_ges_disc.sh` script to download MERRA-2 meteorological data and process with MERRA2WRF.
- Disabled call to WRF-Chem optical driver when using 2011 or 2014 Goddard radiation.
- Bug fixes to GOCART volumetric mean refractive index calculation:
  1. Removed erroneous copying of dust to “other inorganics”, which created spurious extra source of aerosols; and
  2. Removed MSA from refractive index calculation.
- Updated 2014 Goddard Radiation scheme:
  - Rescaled molecular absorption up 50% in shortwave scheme to correct severe underestimation of clear-sky atmospheric absorption. Value adjusted towards RRTMG scheme.
  - Increased CO$_2$ concentration.
  - New lookup tables for 4ICE microphysics.
  - Code changes to support OpenMP.
- Lateral boundary condition bug fix (cherrypicked from WRF 3.7).
- Increase in maximum I/O time levels.
- Added kinetic energy spectra scripts (from NASA DSCALE project).
- Upgraded GSDSU to version 3.5:
  - New global IR emissivity database from University of Wisconsin (Seemann et al. 2008).
  - Modified size boundaries for precipitating and non-precipitating particles (200-µm diameter threshold) for computing microwave/visible/IR optical properties.
  - UV-VIS-IR non-spherical ice scattering database for VISIR simulator from Texas A&M University team (Yang et al. 2013).
- Bug fixes for memory allocation, writing to netCDF output file, and handing MPI aborts.
- Updated GSDSU User Guide.
- Modified LDT to output netCDF4 file for REAL if selected at compile time. Earlier code was hardwired to use classic netCDF3, which made LDT unusable for very large domains (e.g., 5000 × 5000).
2.3 History

- Bug fix to G3 cumulus scheme to prevent division by zero in Himalayas.

- Changed PREP_CHEM_SOURCES to support new 72-level GOCART background data files for actual year and month being processed. The user should edit `prep_chem_sources.inp` and set `gocart_bg.data_type="new"`; create a `gocart_bg_new` subdirectory on the same level as the `gocart_bg` folder containing the community 55-level files; and store new `geos5_met_1MAVG_YYYYMM.nc` and `gmi_merra_oxidants_YYYYMM_1.25x1.nc` files in `gocart_bg_new`, where YYYY is the 4-digit year and MM is the 2-digit month. If PREP_CHEM_SOURCES cannot find both files, it will fall back on `gmi_2006MM.nc` files in `gocart_bg_new`, which are also 72-levels but only exist for 2006.

- Added limited support in UPP for Goddard microphysics schemes. Graupel and hail mixing ratios are now read in. If the NU-WRF 3ICE or 4ICE scheme was used (instead of the community WRF version), then the DBZ field will also be read in. Radiance generation is disabled (use GSDSU for that). Since hail is not in the NCEP GRIB table, hail and graupel mixing ratios are combined. This may lead to problems with the GSD visibility product, but the proper extinction coefficients for hail are unknown.

- Version 7-3.5.1-p4 (Official “Arthur” Patch Release 4)
  - Upgraded LIS to 7.1r. This includes:
    * Noah 3.6, NoahMP 3.6, and CABLE 1.4b LSMs (only Noah 3.6 tested with NU-WRF).
    * Flood and drip irrigation.
    * VIIRS Daily GVF data
    * TRMM 3B42 V7 real time precipitation
    * Gaussian T1534 GFS met forcing data
    * MERRA-2 met forcing data (data only available to select users).
    * Downscaling precipitation (PRISM) (NLDAS-2 only)
    * Should compile now with gfortran.
  - Upgraded LDT to 7.1r. Updates include:
    * Fixes to several metforcing readers that process the forcing terrain-height or elevation fields when processing model parameters.
    * Corrected conserved (budget-bilinear) interp routines. Users may notice slight differences in results.
    * MetTimeDScale and Met forcproc runmodes.
    * Crop, CLM2, Flake, Mosaic, Noah, SiB2, and VIC parameters.
    * TRMM 3B42 V7 real time precipitation.
    * Aquarius L2, GCOMW AMSR2 L3, and SMOS L2 soil moisture observations.
* Simulated GRACE products.

- Upgraded LVT to 7.1r. Updates include:
  * Fixed bug affecting output of summary and final metric files.
  * GCOMW, ASCAT, SMOS, and SMOS L1 Tb observations.
  * MODIS LST and Great Lakes Hydro data.
  * Time lagged computations.

- Upgraded compiler version to Intel 13.1.3.192.

** Version 7-3.5.1-p3 (Official “Arthur” Bug Fix Release 3)

- Updated pleiades.cfg to use new generic SGI MPT module on Pleiades. Older versions of SGI MPT are scheduled for removal on 6 July 2015, and users are advised to switch to the generic module to prevent future breakage when SGI MPT is upgraded and older versions are removed.

  - Fixed netCDF4 chunking and deflation (compression) in GOCART2WRF when writing new chemistry fields to wrfinput and wrfbdy. Also removed unused variables from source code, and reorganized Makefiles.

  - Added new temporalInterpolation GEOS2WRF tool, plus run_temporalInterpolation_merra2_3hr.discover.sh script to automate and interpolate 6-hourly MERRA-2 reanalyses to 3-hourly.

  - Upgraded LIS to 7.0rp3.

  - Added options to suppress dynamic dust emission and to spread dust across multiple near-surface levels.

  - Updated sample batch scripts to check if SGI MPT, Intel MPI, or MVAPICH2 is used and invoke the corresponding appropriate run command when starting a MPI program (mpirun, mpiexec, mpiexec.mpt, or mpiexec.hydra).

  - Modified ARWPOST to read wrfout files from WRF-Chem.

  - Modified PREP_CHEM_SOURCES to support NASA QFED fire emissions.

  - Modified Run_MERRA2.csh to allow user to override base path of MERRA-2 data via environment variable.

  - Added build system option to compile WRF with Allinea MAP profiling libraries (only works on Discover).

** Version 7-3.5.1-p2 (Official “Arthur” Bug Fix Release 2)

- Fixed WRF-Chem GOCART cloud fraction code to run when not using Grell cumulus scheme.

- Updates to LIS MERRA-Land reader code – better handling of missing input.
2.3 History

- Updated Run_MERRA2.csh script to reflect current stream status and to use consistent indentation.
- Added MERRA-2 support in LIS.
- Added run_geos2wrf_merra2_assim.discover.sh script to run GEOS2WRF to process 3-hourly MERRA-2 assimilation data (GEOS-5 output as the model is adjusting towards the MERRA-2 6-hr analyses via IAU). The collection for this data is a bit different than the 6-hr analyses, and it was easier to use GEOS2WRF rather than edit MERRA2WRF. Script only works on Discover, and assumes the user has access to the MERRA-2 GMAO directory on Discover.
- Added bug fix to GEOS2WPS to handle 3-hourly MERRA-2 assimilation data.
- Bug fix for MERRA-Land in LIS (fixes downward shortwave radiation index).
- Allow WRF to run with Noah LSM selection if REAL processed LIS data. Allows user to switch from WRF-LIS coupling to WRF-Noah-initialized-from-LIS without rerunning LIS to produce GRIB files.
- Updated LIS/LDT/LVT Makefiles to reduce number of include directories to search through (cuts down on build time).
- Fixed loops in WRF-Chem convective transport code to skip nv=1 (which is not a valid chemical species), and to loop through all chemical species instead of just gas (back ported from WRF-Chem 3.6.1).
- Added value for variable DELZ_THRESOLD in WRF-Chem plume model. Plume model with stop when plume top changes less than 100 meters over ten minutes.
- Bug fix to PREP_CHEM_SOURCES, resolving conflicting symbols imported from two different modules.

Version 7.3.5.1-p1 (Official “Arthur” Bug Fix Release)

- Modified build configuration files for Discover to check operating system version on computer used for compilation. New default configuration requires SLES 11.3 (same as the new Haswell nodes) and Intel MPI 5. A separate configuration file discover_intel13_sgimpt_sp3.cfg is also added to allow use of SGI MPT. Older Intel MPI 4 and MVAPICH2 configurations are moved to discover_intel13_impi4_sp1.cfg and discover_intel13_mvapich2_sp1.cfg, respectively, and require the older SLES 11.1 operating system.
- Fixed build configuration file pleiades_intel13_impi.cfg to compile on Pleiades with Intel MPI (default configuration uses SGI MPT).
- Merged in LIS 7.0rp2. Bug fixes: Do not reset Zh and Zm for processes with zero tiles; several VIC LSM related changes, and updates for CMAP.
2.3 History

- Merged in LDT 7.0rp2. Bug fixes to mapping between LIS fine Lambert Conformal grid and coarser Lat-Lon parameter grid extents. Also fixes domain extent checks for NLDAS-1 and NLDAS-2 forcing.
- Bug fixes to Goddard 2011 and 2014 radiation schemes. Transmission functions in the CO2, O3, and three water vapor bands with strong absorption are now computed using table look-up. Significantly improves accuracy for pressures less than 10 mb.
- Modified WRF and LIS configure templates to add -xCORE-AVX2 compiler flag options. However, early tests show slower run times compared to -xSSE4.2, so the build cfg files do not currently use them.
- Merged in Weile Wang’s (NASA ARC) modifications to the WRF spectral nudging code. FFT runs significantly faster with Weile’s changes.
- Modified WRF code to fix -DBENCH instrumentation for most parts of the WRF solver.
- Added Python scripts to calculate summary metrics from WRF RSL files for benchmarking.
- Modified MERRA2WRF to add preliminary support for MERRA-2 data files released to select users by the GMAO. “Official” support will not occur until after MERRA-2 is released to the general public.
- Build system fixes for handling CASA preprocessors.
- Updated sample Discover batch scripts for Haswell nodes.

- Version 7-3.5.1 “Arthur” (Official Release)
  - Merged WRF 3.5.1, WPS 3.5.1, PREP_CHEM_SOURCES 1.3.2, RIP 4.6.5, UPP 2.1, and MET 4.1.
  - Merged LIS 7.0rp1, LDT 7rp1, and LVT 7r, and updated LISCONFIG.
  - Merged GSDSU 3.3.3.
  - Modified build system to use netCDF4 with HDF5 compression for all programs that rely on netCDF. Also removed fisreal build option, turning all LIS related compile-time code changes into run-time changes.
  - Added 2014 Goddard radiation package, new Goddard 4ICE microphysics, and new Goddard 3ICE microphysics. Community WRF versions of Goddard microphysics and radiation are now separate options.
  - Added new soil erodibility options to WRF-Chem: MDB, DYN_CLIMO, and DYN.
  - Skin temperature bug fix for restarts when using time-varying sea ice.
2.3 History

- Fixed processing of GFEDv3.1 biomass burning emissions.
- Improved interpolation of GOCART background fields.
- Added support for MERRAero data in addition to the GEOS-5 GO-CART data to use in WRF-Chem using GOCART2WRF utility.
- Updated Vtable.LIS for optional UNGRIB processing of LIS GRIB2 files.
- Revised WRF diagnostic mean and standard deviation calculations to use Welford algorithm (less sensitive to roundoff errors, avoids NaNs when fields vary little with time).
- Added on-line diagnostics: precipitable water, liquid water path, ice water path, cloud liquid water path, cloud ice water path, rain water path, frozen precipitation water path, time-averaged integrated water vapor transport vector, and freezing level.
- Added spatial subsetting option for GEOS2WPS program in GEOS2WRF.
- Added support for CASA climatological CO$_2$ tracers in WRF-Chem, including preprocessors.
- Removed SZIP library build dependencies.
- Updated sample batch scripts for Pleiades and Discover.
- Temporarily dropped support for gfortran compiler.
- Upgraded to SGI MPT 2.11r13 on Pleiades with Intel compilers.
- Added MVAPICH2 support on Discover with Intel compilers.

• Version 6-3.4.1-p2 (Official Bug Patch Release)
  - Merged in LIS 6.1rp7 updates, including fixes to latitude of NARR forcing, and bug fixes to WRFout reader.
  - Bug fixes to PREP, CHEM, SOURCES (memory allocation and namelist initialization).
  - Improved error checking for NaNs.
  - Source code updates to allow compilation on OS X with gfortran and OpenMPI.
  - Bug fix to MET/pcp,combine ensuring closure of files.
  - Fixed WRF bug for changing restart dump interval when simulation is itself restarted.
  - Build system improvements for LIS, UPP, and GSDSU.
  - Reorganized sample batch scripts into Discover/SLURM and Pleiades/PBS versions.

• Version 6-3.4.1-p1 (Official Bug Patch Release)
  - LIS bug fix (for directory creation) to allow use with SGI MPT.
2.3 History

- Updated optimization flags, targeting Westmere or newer Intel hardware.
- Upgraded to Intel MPI 4.0.3.008 on Discover, and to SGI MPT 2.08r7 on Pleiades.
- Build system tweaks for WRF and LIS to reduce compile times.
- Build system fix for cleaning GSDSU.

- Version 6-3.4.1 (Official Release)
  - Merged in WRF 3.4.1, WPS 3.4.1, UPP 2.0, and multiple MET 4.0 bug patches.
  - Merged in LIS 6.1rp6.
  - Merged in GSDSU 3.0.
  - Overhauled WRF-LIS coupling. Added lisreal build option to compile WRF and REAL with special logic supporting LIS. Added &lis block to namelist.input to allow REAL to process LIS netCDF output file and update wrfinput accordingly.
  - Added support for compiling LIS as standalone executable. Added wrfout plugin to use WRF netCDF files as forcing for LIS. Added deep soil lapse rate option to adjust deep soil temperature in high terrain. Added dynamic deep soil temperature option to change as function of time-lagged skin temperature.
  - Updated Goddard microphysics to add variables rainncv_sepa and rainnc_sepa, and added multiple bug fixes.
  - Bug fixes to UPP for lightning threat product.
  - Several bug fixes to build system to compile GSDSU, LIS, and WRF on Pleiades. Also better handles netCDF and ESMF paths for LVT.
  - Added script support for plotting composite reflectivity, surface reflectivity, and skin temperature with RIP4.
  - Upgraded to Intel 13 compilers on Discover and Pleiades.

- Version 5-3.4 (Internal Beta Release)
  - Merged WRF 3.4, WPS 3.4, RIP 4.6.3, UPP 1.1, MET 4.0 (with patches through 29 June 2012), and PREP_CHEM_SOURCES 1.2_10apr2012.
  - Merged in GEOS2WRF 2.0.
  - Bug fix to CONVERT_EMISS to be compatible with PREP_CHEM_SOURCES.
  - Bug fixes to GOCART2WRF 2.0.
  - Added user defined tuning factors for GOCART dust emissions. Bug fixes for aerosols and AFWA GOCART dust emissions.
  - LIS bug fix for porosity: values from Noah and CLM2 LSMs now passed back to WRF and used in GOCART dust emissions.
2.3 History

– Overhauled LISCONFIG program to process GEOGRID output instead of METGRID output.
– Upgraded compilers to Intel 12.1 on Discover and Pleiades.

• Version 4-3.3.1 (Official Release)
  – Merged in WRF 3.3.1, WPS 3.3.1, ARWpost 3.1, RIP 4.6.2, MET 3.1, UPP 1.0, and PREP_CHEM_SOURCES v2.04aug2011 updates. Removed WPP.
  – Added GSDSU V3BETA.
  – Added WPS map projection support to PREP_CHEM_SOURCES, and added PLOT_CHEM utility to display output. Modified CONVERT_EMISS to support new 72-level GOCART background files with improved vertical positioning. Upgraded GOCART2WRF to version 2.0 (supporting GEOS-5 netCDF4 files), and removed support for old GEOS-4 netCDF3 files. Added porosity calculation as function of USGS land use for GOCART dust emissions.
  – Updated GEOS2WRF to read specific humidity from GEOS-5 HDF4 file and convert to relative humidity. Added bug fixes for LANDSEA and 2-meter relative humidity. Added scripting to ease use.
  – Added MERRA2WRF 2.0 to process HDF4 and netCDF MERRA files from the NASA Goddard Earth Sciences Data Information Services Center (GES DISC).
  – Fixed bug in RIP4 preventing processing of wrfout files from WRF-Chem.
  – Modifed build system to optionally compile WRF-Chem with KPP.
  – Added sample batch scripts and input files for running GEOGRID, UNGRIB, METGRID, REAL, WRF, and RIP on Discover.

• Version 3-3.2.1 (Official Release)
  – Merged in WRF 3.2.1 and WPS 3.2.1. Added PREP_CHEM_SOURCES.
  – Merged in LIS 6.1rp1. Added LISCONFIG tool to customized lis.config file based on WRF domain and map projection settings. Added LVT.
  – Fixed GOCART2WRF bug triggered when WRF and GOCART pressures were identical, and added support for inner-nest domains. Added fromGEOS5_to_GEOS4 utility to convert on-line GEOS-5 GOCART data to off-line GEOS-4 GOCART format for GOCART2WRF.
  – Updated Goddard radiation and aerosol coupling component with bug fixes to prevent negative effective radii and division-by-zero, and handle spurious negative mixing ratios.
  – Added Goddard Microphysics-GOCART aerosol coupling.
2.3 History

- WRF-Chem updates: Added capability of estimating SOA from biogenic terpene emissions, including three new variables: \( e_{\text{terp}} \), \( e_{\text{api}} \), \( e_{\text{lim}} \). Linked MEGAN2 biogenic emissions scheme, GOCART dry deposition scheme, and various optical property schemes to GOCART aerosols. Linked RADM2 chemistry to GOCARTRADM2 option. Added namelist options for radiation-aerosol and microphysics-aerosol coupling (Goddard and GOCART). Added bio_emiss_soa namelist option to toggle emission conversion to SOA. Removed a number of chemistry variables from wrfout file.

- GSDSU updates: Added GOCART input options to WRF input, GCE-SBM 3D option, SBM moment output, GrADS control file output, and Morrison two-moment support, plus bug fixes.

- Build system updates: Set incremental building as the default option (instead of performing a complete rebuild). Updated Discover configuration to use Intel MPI 4.0.1.007-beta, and added support for Pleiades. Added automatic detection of default configuration file on Discover and Pleiades. Add compilation of CONVERT_EMISS when 'chem' target is selected.

- Version 2-3.2.1 (Official Release)
  - Merged in MET bug fixes through 15 Feb 2011.
  - Updated LIS to version 6.0rp6.
  - WRF-LIS coupling: LIS export data not longer overwrites WRF data at water-points. Run-time checks for surface physics scheme setting in wrfinput disabled. REAL now built in special mode to generate consistent initial conditions for coupled WRF-LIS runs.
  - Goddard radiation and GOCART are coupled when running with WRF-Chem.
  - Severe weather diagnostics added to WPP for applicable physics schemes. Diagnostics include: Maximum 10-meter wind speed, column mean vertical velocity, max column integrated graupel, maximum lightning threat, derived radar reflectivity, precipitation accumulation for a given time window, and convective precipitation accumulation.
  - Changed MERRA2WRF to output relative humidity rather than specific humidity due to bug in REAL. Added improved error checking when calling the HDF4 library. Also, batch script for running on Discover changed to gracefully handle back PBS charge codes.
  - Changed GOCART2WRF to calculate correct tendencies at the final time level, and adds error checking when calling the netCDF library.
  - Wrote unified build system to compile all components of NU-WRF, targeting NASA’s NCCS Discover system.
2.3 History

- Version 1.3.1.1 (Official Release)
  - Includes WRF 3.1.1, WPS 3.1.1, ARWpost 2.1, RIP 4.5, MET 2.0, WPP.
  - Includes LIS 6.0rp1, SDSU, SST2WRF 1.0, GEOS2WRF 1.0, MERRA2WRF 1.0
  - Includes simple build script for WRF on Discover.
  - Contains new NASA microphysics and radiation.
  - LIS integrated as WRF component.
  - Added severe weather diagnostics from NASA MSFC SPoRT.
3 Obtaining Software

3.1 Software Usage Agreement

The release of NU-WRF software is subject to NASA legal review, and requires users to sign a Software Usage Agreement. Toshi Matsui (toshihisa.matsui-1@nasa.gov) and Eric Kemp (eric.kemp@nasa.gov) are the points of contact for discussing and processing requests for the NU-WRF software.

There are three broad categories for software release:

1. **US Government – Interagency Release.** A representative of a US government agency should initiate contact and provide the following information:
   
   (a) The name and division of the government agency
   (b) The name of the Recipient of the NU-WRF source code
   (c) The Recipient’s title/position
   (d) The Recipient’s address
   (e) The Recipient’s phone and FAX number
   (f) The Recipient’s e-mail address

2. **US Government – Project Release under a Contract.** If a group working under contract or grant for a US government agency requires the NU-WRF source code for the performance of said contract or grant, then a representative should initiate contact and provide a *copy of the grant or contract cover page*. Information should include the following:

   (a) The name and division of the government agency
   (b) The name of the Recipient of the NU-WRF source code
   (c) The Recipient’s title/position
   (d) The Recipient’s address
   (e) The Recipient’s phone and FAX number
   (f) The Recipient’s e-mail address
   (g) The contract or grant number
   (h) The name of the Contracting Officer
   (i) The Contracting Officer’s phone number
   (j) The Contracting Officer’s e-mail address

3. **All Others.** Those who do not fall under the above two categories but who wish to use NU-WRF software should initiate contact to discuss possibilities for collaborating. Note, however, that NASA cannot accept all requests due to legal constraints.
3.2 Tar File

The Recipient will be provided a compressed tar file containing the entire NU-WRF source code distribution. (NU-WRF project members have access to tar files pre-staged on the NASA Discover and Pleaides supercomputers.) Two variants are available: gzip compressed (nu-wrf_v8p1-wrf371-lis71rp7.tgz) and bzip2 compressed (nu-wrf_v8p1-wrf371-lis71rp7.tar.bz2). Bzip2 compression generates slightly smaller files but can take considerably longer to decompress.

To untar, type either `tar -zxvf nu-wrf_v8p1-wrf371-lis71rp7.tgz` or `bunzip2 nu-wrf_v8p1-wrf371-lis71rp7.tar.bz2 ; tar -xvf nu-wrf_v8p1-wrf371-lis71rp7.tar`. A `nuwrf_v8p1-wrf371-lis71rp7` directory should be created.

3.3 Subversion Repository

NU-WRF developers have the alternative of pulling code directly from the Subversion (SVN) repository. (See [http://svnbook.red-bean.com](http://svnbook.red-bean.com) for detailed information on using SVN.) This approach requires several set-up steps to comply with NASA security requirements.

First, the developer will require an account on the NASA Center for Climate Simulations (NCCS) Discover supercomputer. The developer should refer to the NCCS website for details ([http://www.nccs.nasa.gov/user_info/new_user](http://www.nccs.nasa.gov/user_info/new_user)).

Second, the developer should contact repository manager Eric Kemp (eric.kemp@nasa.gov) and provide (1) the NCCS username, and (2) the project being worked on. Confirmation from the NU-WRF Principal Investigator may be required before access is granted to the repository.

Third, the developer should create a SSH public key unless they have already created a key on Discover. To create a key, run `ssh-keygen -t rsa` on Discover. Note that RSA encryption is required.

Fourth, the developer should upload the ssh public key to the NCCS Progress repository server (see [https://progress.nccs.nasa.gov/keyupload](https://progress.nccs.nasa.gov/keyupload)). Note that it will take a few minutes for the uploaded public key to be recognized by the server.

Fifth, the developer should add a virtual host entry on Discover. Open or create the file `$HOME/.ssh/config` and add the following entry:

```
Host progressdirect
 Hostname progress.nccs.nasa.gov
 Port 22223
```

Once set-up, the developer can export the source code using the following command on Discover:

```
svn export svn+ssh://progressdirect/svn/nu-wrf/code/tags/releases/v8p1-wrf371-lis71rp7
```
3.4 Directory Structure

The source code directory structure is as follows:

- The ARWpost/, GSBU/, ldt/, lvt/, MET/, RIP4/, UPP/, utils/, WPS/, and WRFV3/ folders contain the source codes of the components summarized in Section 2.1.

- The docs/ folder contains documentation on the different NU-WRF components.

- The oldcfg/ folder contains retired build configuration files for different platforms, compilers, and libraries. These are included to aid users in porting NU-WRF to a non-supported configuration.

- The scripts/ folder contains sample batch scripts for running a number of NU-WRF component programs on the NASA Discover and Pleiades supercomputers (in the discover/ and pleiades/ subfolders, respectively). A fetch_data/ directory also exists with a script to download SST data from NASA SPoRT. Sample input files for RIP4 are stored in the rip/ subfolder. A script for creating new tags from the main development branch in the SVN repository is also included in the devel/ subfolder.

- The defaults/ folder contains sample namelist and other input files for NU-WRF with default physics and other settings.

- The testcases/ folder contains sample namelist and other input files for NU-WRF for different configurations (simple WRF, WRF with LIS, WRF-Chem, and WRF-KPP).

The main directory also includes Bash scripts for the unified NU-WRF build system (discussed in Section 4.3) and a CHANGELOG.TXT file summarizing the evolution of the overall modeling system.
4 Building Software

4.1 Compilers

The NU-WRF source code requires Fortran 90/2003, C, and C++ compilers. The current release officially supports Intel compilers 15.0.3.187 (ifort, icc, and icpc) on Discover and Pleiades. GCC 5.3 is also required for library compatibility for icpc.

4.2 External Libraries and Tools

A large number of third party libraries must be installed before building NU-WRF. Except as noted, the libraries must be compiled using the same compilers as NU-WRF, and it is highly recommended that static library files be created and linked rather than shared object. The list is as follows:

- A MPI library. (By default, NU-WRF uses SGI MPT 2.12 on Discover and the latest SGI MPT on Pleiades. Intel MPI 5.1.2.150 should be used on Discover if using legacy Sandy Bridge compute nodes.)
- BUFRLIB 10.2.3. (Newer versions cannot be used due to incompatibilities with MET.)
- ESMF 6.3.0rp1 compiled with MPI support.
- FREETYPE 2.6.5.
- FLEX (can use precompiled system binary on Linux).
- G2CLIB 1.6.0.
- GHOSTSCRIPT 8.11.
- GRIB_API 1.17.0.
- GSL 2.2.1.
- HDF4 4.2.12.
- HDF5 1.8.17.
- HDF-EOS 2.19v1.00.
- JASPER 1.900.1.
- JPEG 6b.
- LIBPNG 1.2.56. (The newer 1.5.* versions cannot be used due to incompatibilities with WPS.)
4.3 Build System

- NCAR Graphics 6.0.0. (Newer versions of NCAR Graphics require compilation and linking to the CAIRO library, which is currently only supported by MET.)

- NETCDF 4.4.1 (C library), NETCDF-Fortran 4.4.4, and NETCDF-CXX 4.2 (C++ library), built with HDF5 compression. (Newer versions of the netCDF C++ API cannot be used due to incompatibilities with MET 5.1.)

- PIXMAN 0.34.0.

- YACC (can use preinstalled version on Linux).

- ZLIB 1.2.8.

(Support for Parallel-NETCDF and/or MPI-IO may be added in the near future.)

In addition to the above libraries, NU-WRF requires Perl, Python, Bash, Tcsh, GNU Make, Sed, Awk, M4, and the UNIX “uname” command to be available on the computer.

4.3 Build System

Each component of the NU-WRF modeling system has a unique compilation mechanism, ranging from simple Makefiles to sophisticated Perl and shell scripts. To make it easier for the user to create desired executables and to more easily resolve dependencies between components, NU-WRF includes a set of high level “wrapper” scripts for compilation. Each wrapper script is designed to be used by a common master script, and is customized to directly manage the component-specific build mechanism and inject appropriate configuration settings into that build mechanism. For example, the wrapper script for WPS will modify the configure.wps generated by configure [see Chapter 2 of NCAR 2016] to update several library paths; the modified configure.wps is then used by the WPS compile script.

The build system allows users to specify configure options for the AR-WPOST, RIP4, UPP, WPS, and WRFV3 components, as well as template Makefile names for the CASA2WRF, GEOS2WRF, GOCART2WRF, GSDSU, LISCONFIG, LVT, MET, NDVIBARENESS4WRF, PREP.CHEM.SOURCES, and SST2WRF components. These options are stored in the build configuration file (with names like discover.cfg and pleiades.cfg). This approach should aid in porting NU-WRF to new compilers, MPI implementations, and/or operating systems, a process discussed more fully in Appendix B.

With this system, the user can build executables by invoking a single Bash driver script called build.sh located in the top-level directory. This script accepts three types of command-line arguments:

- Configuration. The --config flag followed by the name of a configuration file specifying critical environment variables (e.g., the path to the netCDF
4.3 Build System

library). Current configuration files are included in the top-level directory: discover.cfg, discover_intel15_impi5_sp3.cfg, and pleiades.cfg. Users may develop their own configuration file to customize settings. If the configuration arguments are skipped, build.sh will default to either discover.cfg or pleiades.cfg based on the local environment, or exit if the software is on an unrecognized computer. Note that the configurations for Discover will check for specific operating system versions and abort if they are not found – this is to ensure consistency with the operating system on the desired compute nodes (currently SLES 11.3).

- **Options.** The user may specify cleanfirst, debug, and/or nest=n where \(n\) is an integer ranging from 1 to 3.
  - The cleanfirst option will cause the build system to “clean” a target (delete object files and static libraries) before starting compilation.
  - The debug option forces the WRFV3 or WPS build subsystems to use alternative compilation flags set in the configuration file (e.g., for disabling optimization and turning on run-time array bounds checking). This option is currently ignored by other NU-WRF components.
  - The nest=n option specifies compiling WRF with basic nesting \((n=1)\), preset-moves nesting \((n=2)\), or vortex-tracking nesting \((n=3)\). Basic nesting is assumed by default. Note that WRF cannot be run coupled to LIS if preset-moves or vortex-tracking nesting is used. Similarly, WRF-Chem only runs with basic nesting.
  - The profileMAP option links WRF against special libraries required by the Allinea MAP commercial profiling tool (see [http://www.allinea.com](http://www.allinea.com)). This only works on Discover, and is only intended to perform special profiling runs for subsequent code modification and runtime improvement.

- **Targets.** The user can compile all or select executables with variations on chemistry support, as well as delete all executables, object files, and static libraries. The recognized targets are:
  - The all target compiles all executables without WRF-Chem.
  - The allchem target compiles all executables with WRF-Chem but without the Kinetic Pre-Processor.
  - The allclean target deletes all executables, object files, and static libraries.
  - The allkpp target compiles all executables with WRF-Chem and including the Kinetic Pre-Processor.
  - The arwpost target compiles executables in the ARWpost/ directory.
  - The casa2wrf target compiles executables in the utils/casa2wrf/ directory.
4.3 Build System

- The **chem** target is an alias for **wrfchem**.
- The **geos2wrf** target compiles executables in the `utils/geos2wrf_2/` directory (both GEOS2WRF and MERRA2WRF).
- The **gocart2wrf** target compiles executables in the `utils/gocart2wrf_2/` directory.
- The **gsdsu** target compiles executables in the `GSDSU/` directory.
- The **kpp** target is an alias for **wrfkpp**.
- The **ldt** target compiles executables in the `ldt/` directory.
- The **lis** target compiles LIS in uncoupled mode in the `WRFV3/lis/make/` directory.
- The **lisconfig** target compiles executables in the `utils/lisconfig/` directory.
- The **lvt** target compiles executables in the `lvt/` directory.
- The **merra2wrf** target is an alias for **geos2wrf**.
- The **met** target compiles executables in the `MET/` directory.
- The **ndviBarenness4Wrf** target compiles executables in the `utils/ndviBarenness4Wrf` directory.
- The **plot_chem** target compiles executables in the `utils/plot_chem/` directory.
- The **prep_chem_sources** target compiles executables in the `utils/prep_chem_sources/` directory.
- The **rip** target compiles executables in the `RIP4/` directory.
- The **sst2wrf** target compiles executables in the `utils/sst2wrf/` directory.
- The **upp** target compiles executables in the `UPP/` directory.
- The **wps** target compiles executables in the `WPS/` directory.
- The **wrf** target compiles executables in the `WRFV3/` directory – except for LIS and CONV_EMISS – without WRF-Chem support.
- The **wrfchem** target compiles executables in the `WRFV3/` directory – except for LIS – with WRF-Chem support but without the Kinetic Pre-Processor. The compiled executables include CONV_EMISS.
- The **wrfkpp** target compiles executables in the `WRFV3/` directory – except for LIS – with WRF-Chem and Kinetic Pre-Processor support. The compiled executables include CONV_EMISS.

One complication addressed by the build system is that the WPS and UPP components are dependent on libraries and object files in the `WRFV3/` directory. To account for this, the `WRFV3/` directory will be automatically cleaned and the WRF component compiled if necessary for WPS and/or UPP, even if the **wrf** target is not listed on the command line.
A second complication is that the coupling of LIS to WRF requires linking the WRFV3/ code to the ESMF and ZLIB libraries. As a result, the configure.wrf file [see Chapter 2 of NCAR (2016)] is modified to link against these libraries. A similar modification occurs for UPP. (No modification is needed for WPS as long as WPS is compiled with GRIB2 support.)

The most straightforward way to compile the full NU-WRF system on Discover or Pleiades is to type ./build.sh all in the top level directory. If chemistry is required, the command is ./build.sh allchem (./build.sh allkpp if KPP-enabled chemistry is needed). To fully clean the entire system, run ./build.sh allclean. (Recall that the build script will automatically select discover.cfg or pleiades.cfg, as appropriate, if it detects the software is being compiled on Discover or Pleiades.) To explicitly specify the configuration file, type e.g., ./build.sh --config discover.cfg all.

The user can selectively build components by listing specific targets. For example, to build the WRF model without chemistry along with WPS and UPP, type ./build.sh wrf wps upp.
5 Front-End Workflows

In this section we will summarize several “front-end” workflows involving the main NU-WRF model and different pre-processors. (Post-processing is discussed in Section 6). The intent is to illustrate the roles of the pre-processors within the NU-WRF system, and to show several different configurations possible with NU-WRF (e.g., advanced land surface initialization, aerosol coupling, and CO$_2$ tracer simulation).

5.1 Basic Workflow

This is the simplest approach to running simulations with NU-WRF. Neither chemistry nor advanced land surface initialization are used, so the user should compile NU-WRF with ./build.sh wrf wps.

- **WPS**: The user must edit a namelist.wps file to customize the WRF domains, set the start and end dates, set the file formats, and provide information on desired terrestrial data and file prefixes. Sample namelist.wps files can be found in the WPS/, defaults/, and testcases/ directory. The user must then run the following programs [see Chapter 3 of NCAR (2016)].

  - GEOGRID. This program will interpolate static and climatological terrestrial data (land use, albedo, vegetation greenness, etc) to each WRF grid. The user should use the GEOGRID.TBL.ARW located in the WPS/geogrid/ directory to specify interpolation options for each dataset selected in namelist.wps. (Alternative GEOGRID.TBL files are available for chemistry cases.) The user is also responsible for obtaining the geog/ dataset from NCAR for processing by GEOGRID. Sample run scripts are available in the scripts/ directory.

  - link_grib.csh. This script is used to create symbolic links to the GRIB or GRIB2 files that are to be processed. The links follow a particular naming convention (GRIFFILE.AAA, GRIFFILE.AAB, ..., GRIFFILE.ZZZ) that is required by UNGRIB.

  - UNGRIB. This program will read GRIB or GRIB2 files with dynamic meteorological and dynamic terrestrial data (soil moisture, soil temperature, sea surface temperature, sea ice, etc) and write specific fields in WPS intermediate format. The user must select an appropriate Vtable file in WPS/ungrib/Variable_Tables/ to specify the fields to be extracted.

  - METGRID. This program will horizontally interpolate the output from UNGRIB to the WRF domains, and combine them with the output from GEOGRID. The user must select the METGRID.TBL.ARW file to specify the interpolation methods used by METGRID for each field.
5.2 Land Surface Initialization and LIS Coupling

- **REAL.** This program will vertically interpolate the METGRID output to the WRF grid, and create initial and lateral boundary condition files. REAL is described in Chapters 4 and 5 of [NCAR (2016)](https://www.ucar.edu/). The user must edit a namelist.input file to specify the WRF domains, start and end times, and WRF physics configurations for REAL. *A standard WRF land surface model should be selected for this workflow. No chemistry options can be selected.* Sample namelist.input files are available in the testcases/ directory.

- **WRF.** This program will perform a numerical weather prediction simulation using the data from REAL. User need to change the namelist.input file for specific run cases. A sample namelist.input file can be found in the WRFV3/run/ directory and in testcases/. WRF is described in Chapter 5 of [NCAR (2016)](https://www.ucar.edu/). In addition to the normal WRF physics options, the user can specify the new Goddard 3ICE or 4ICE microphysics (mp.physics=55 or 56), and the new Goddard 2011 or 2014 radiation schemes (ra_lw.physics=55 or 56 for longwave, and ra_sw.physics=55 or 56 for shortwave), all without aerosol coupling. Note that the 2014 radiation scheme requires users to create a symbolic link to the WRFV3/GODDARDRAD_SSLUT/ directory, and put that link in the directory where the model is run.

A feature added to NU-WRF is the calculation of mean integrated vapor transport. The user may adjust the time-averaging period for this diagnostic by changing the IVT_INTERVAL flag in the &time_control block of namelist.input. A value of 0 indicates instantaneous values will be output as the “means”, while positive values indicates averaging time periods in minutes.

---

**Basic WRF workflow**

WRF Preprocessing System

WPS

GEOGRID

UNGRIB

WRF-ARW

(Advanced Research WRF)

---

**5.2 Land Surface Initialization and LIS Coupling**

This is a more advanced approach to running simulations with NU-WRF. Instead of using land surface fields interpolated from a coarser model or reanalysis, a custom-made land surface state is created by LIS on the same grid and with the same terrestrial data and land surface physics as WRF. WRF will then
5.2 Land Surface Initialization and LIS Coupling

call LIS on each advective time step, providing atmospheric forcing data and receiving land surface data (fluxes, albedo, etc) in return.

For simplicity, this workflow uses no chemistry, so the user should compile NU-WRF with 

```bash
./build.sh wrf wps ldt lisconfig
```

However, an advanced user can combine this workflow with one of the chemistry workflows described further down; in that case, the user should replace the `wrf` target with `chem` (or with `kpp` if using KPP-based chemistry).

- **WPS.** These steps are identical to those WPS steps in Section 5.1.

- **LISCONFIG.** The user must provide a `ldt.config` file (used by LDT) and a `lis.config` file (used by LIS). The LISCONFIG software will read the `namelist.wps` file and the netCDF4 output files from GEOGRID, and copy the WRF grid information to the two configuration files. LISCONFIG is divided into two executables: `lisWrfDomain` (a Fortran compiled program found in `utils/lisconfig/bin`) and `lisWrfDomain.py` (a Python wrapper script found in `utils/lisconfig/scripts`).

The software can be run as 

```bash
./lisWrfDomain.py DOMAINPROG LISCONFIG LDTCONFIG WPSDIR
```

where `DOMAINPROG` is the path to `lisWrfDomain`, `LISCONFIG` is the path to `lis.config`, `LDTCONFIG` is the path to `ldt.config`, and `WPSDIR` is the directory containing `namelist.wps` and the GEOGRID netCDF4 output files.

Sample configuration files are provided in `defaults/` and in `testcases/`.

- **LDT.** The user must further customize the `ldt.config` file and a separate parameter attributes file to specify the static and climatological terrestrial data to be processed in “LSM parameter processing mode” [see NASA (2015a)]. For NU-WRF Version 8, the following settings are recommended/supported:

  - Noah.3.6 is the recommended land surface model (Noah 3.3, 3.2, or 2.7.1 can also be used);
  - MODIS is the recommended land use dataset (UMD and USGS are also supported);
  - SRTM30 is the recommended terrain elevation dataset (GTOPO30 is also supported);
  - STATSGOFAO is the recommended soil texture dataset;
  - NCEP monthly climatological albedo and and max snow free albedo are recommended;
  - NCEP monthly climatological and maximum/minimum vegetation greenness are recommended;
  - NCEP slope type is recommended;
  - Use of the “slope-aspect correction” is recommended to improve soil moisture spin-ups; and
5.2 Land Surface Initialization and LIS Coupling

- ISLSCP1 deep soil temperature with terrain lapse-rate correction is recommended (not using the lapse rate correction could result in warm biases in high terrain).

- **LIS.** The user must further customize *lis.config* for a “retrospective” run. This includes specifying the start and end dates of the “spin-up” simulation, identifying the LDT datasets, specifying the land surface model, and identifying the atmospheric forcing datasets. The user must also customize a forcing variables list file compatible with the forcing dataset, and a model output attributes file. All these files are described in more detail in [NASA](2015b). (Recall that only Noah 2.7.1, Noah 3.2, Noah 3.3, and Noah 3.6 can be used with NU-WRF). Sample forcing variable and output attribute files are provided in the *testcases/* directory.

- **LDT.** After running LIS, it is necessary to rerun LDT in “NUWRF preprocessing for real” mode. This requires modifications to *ldt.config* to specify the static output file from LDT and the dynamic output file from LIS. Fields from both will be combined and written to a new netCDF output file for use by REAL. Sample files are given in *testcases/*.

- **REAL.** REAL is run similarly to the configuration in Section 5.1 except that it also reads the static and dynamic land surface data collected by LDT. For this to work, the *namelist.input* file must include an additional namelist block:

```
&lis
   lis_landcover_type = 2,
   lis_filename = "lis4real_input.d01.nc"
/
```

Here *lis_landcover_type* specifies the land use system used with LIS and LDT (1 = USGS, 2 = MODIS, 3 = UMD), and *lis_filename* is an array of character strings specifying the combined LDT/LIS files for each WRF domain.

In addition, the user must specify LIS as the land surface model selection with WRF (sf_surface_physics=55).

The resulting initial and lateral boundary conditions will replace the land surface fields from UNGRIB with those from LDT/LIS.

- **WRF with LIS.** Running WRF in this case is similar to the basic case in Section 5.1 except that WRF will also read the *lis.config* file and the LIS restart files that were produced during the “retrospective” run. The user must modify *lis.config* to run in “WRF coupling mode”, and specify *forcing_variables_wrfcplmode.txt* as the forcing variables list file. The start mode must also be changed to “restart”, and the time step for each LIS domain must match that used with WRF (specified in *namelist.input*).
5.3 Use of GEOS-5 Meteorological Data

One source of initial and lateral boundary conditions is NASA’s GEOS-5 global model (Rienecker et al., 2008). A number of dataset options exist from GEOS-5, including daily near-real-time simulations (see http://gmao.gsfc.nasa.gov/products), and archived MERRA (Rienecker et al., 2011) and MERRA-2 (Bosilovich et al., 2015) reanalyses (both available from http://disc.sci.gsfc.nasa.gov). GEOS-5 can provide not just meteorological fields (temperature, pressure, wind, and moisture), but also aerosol fields due to the use of the GOCART aerosol module (Chin et al., 2002).

There are several challenges to using GEOS-5 data. First, the GEOS-5 land surface data cannot be used to initialize WRF, due to fundamental differences in the GEOS-5 Catchment LSM (Koster et al., 2000) and those in WRF. Users are therefore advised to use the GEOS-5 data in a workflow that also includes WRF-LIS (see Section 5.2 above). Second, GEOS-5 aerosol data cannot currently be handled by WPS and REAL, as these tools were designed for meteorological fields (temperature, pressure, wind, and moisture). Processing these aerosol fields requires a special workflow described in Section 5.5.

Remaining issues involve the format and organization of the GEOS-5 data. GEOS-5 writes output in netCDF (and historically HDF4 and HDFEOS2) instead of GRIB or GRIB2; GEOS-5 allows user-specification of variables and variable names for different output files, leading to wide variations between simulations; and GEOS-5 often does not output all the variables expected by WPS. To address these issues, special preprocessing software has been devel-
5.3 Use of GEOS-5 Meteorological Data

GEOS2WRF (a collection of utilities designed for customized processing of GEOS-5 data, including derivation of missing variables), and MERRA2WRF (a monolithic program customized to process 6-hourly MERRA and MERRA-2 reanalyses).

5.3.1 GEOS2WRF

GEOS2WRF can be broken down into four main sub-groups:

- **Front-end conversion.**
  - **GEOS2WPS.** A front end converter that can read HDF4, netCDF3, and netCDF4 files with GEOS-5 data. A `namelist.geos2wps` file is read in as input, and must be customized to list the location, name, and format of the GEOS-5 file; the names of the coordinate arrays in the GEOS-5 file; number of time slices, the indices of the slices, valid times and forecast hours; and the number of variables to process, along with their names, ranks, input and output names, units, and descriptions. This program takes the place of UNGRIB. The output from GEOS2WPS are written in WPS intermediate format [see Chapter 3 of NCAR (2016)], with the filename convention `$VARNAME-$LEVELTYPE:$YYYY-$MM-$DD-$HH`, where $VARNAME$ is the variable name, $LEVELTYPE$ is a string describing the type of level the data are on, $YYYY$ is the 4-digit year, $MM$ is the 2-digit month, $DD$ is the 2-digit day, and $HH$ is the 2-digit hour. Some example output file names are:

  
  - `TT_MODEL_LEVEL:2009-08-25_00` # Temperature on model levels
  - `PSFC_GROUND_LEVEL:2009-08-25_00` # Surface pressure
  - `PMSL_MEAN_SEA_LEVEL:2009-08-25_00` # Mean sea level pressure
  - `VV_10M_ABOVE_GROUND_LEVEL:2009-08-25_00` # 10-meter V winds

  The $VARNAME$s (TT, PSFC, PMSL, and VV above) are listed in `namelist.geos2wps`, and can be customized by the user; however, they must match the values in the `METGRID.TBL` look-up file used by METGRID for those variables to be processed by WPS. (Intermediate variables used to derive other variables for WPS do not have this naming restriction.)

- **Temporal interpolation.**
  - **temporalInterpolation.** This program takes WPS intermediate file format data and linearly interpolate in time. This can be used, for example, to interpolate 6-hourly MERRA-2 analyses at 00Z, 06Z, 12Z, and 18Z to 3-hourly intervals for initializing WRF at 03Z, 09Z,
5.3 Use of GEOS-5 Meteorological Data

15Z, or 21Z. Only a single, user-specified variable will be processed during a particular program invocation, and other variables in the input data files will be ignored. A `namelist.temporalInterpolation` file is used to specify the variable, input and output data files.

- **Variable-derivation.** Multiple tools for deriving missing variables required by WRF from existing variables. These should be used on an as-needed basis depending on the contents of the GEOS-5 files. Current programs that are in this category are:

  - **createSOILHGT.** A utility that reads in a WPS file with surface geopotential, and calculates the surface terrain field. The output WPS file will be named `SOILHGT_GROUND_LEVEL:YYYY-MM-DD_HH`. A `namelist.createSOILHGT` file is also used as input.
  
  - **createHGT.** A utility that reads in a WPS file with model layer pressure thicknesses, model layer temperatures, model layer specific humidity, and the model terrain field, and derives the geopotential heights on the GEOS-5 model levels. The output WPS files will be named `HGT_MODEL_LEVEL:YYYY-MM-DD_HH`. A `namelist.createHGT` file is also used as input. *This program is not needed when processing isobaric levels.*

  - **createLANDSEA.** A utility that reads in a WPS file with “lake fraction” and “ocean fraction” and derives a land-sea mask. The output WPS files will be named `LANDSEA_GROUND_LEVEL:YYYY-MM-DD_HH`. A `namelist.createLANDSEA` file is also used as input.

  - **createPRESSURE.** A utility that reads in a WPS file with model layer pressure thicknesses, and calculates the (mid-layer) pressures. The output WPS files will be named `PRESSURE_MODEL_LEVEL:YYYY-MM-DD_HH`. A `namelist.createPRESSURE` file is also used as input. *This program is not needed when processing isobaric levels.*

  - **createRH.** A utility that reads in a WPS file with either model or isobaric level temperatures, specific humidity, and pressure, plus optional surface pressure, 2-meter temperature, and 2-meter specific humidity, and derives relative humidity on those levels. The output WPS files will have prefixes of `RH_2M_ABOVE_GROUND_LEVEL`, `RHMODEL_LEVEL`, and/or `RH_ISOBARIC_LEVEL`, and will end with the familiar `$YYYY-MM-DD_HH` string. A `namelist.createRH` file is also used as input. *This program is recommended* because some versions of REAL do not correctly interpolate specific humidity, and because the WRF definition of RH is strictly w.r.t. liquid while some versions of GEOS-5 output a weighted average of RT w.r.t. liquid and ice that is a function of temperature.

- **Extrapolation.**
5.3 Use of GEOS-5 Meteorological Data

- **extrapIsobaric.** A utility that reads in a WPS file with geopotential height, temperature, relative humidity, U and V winds all on isobaric levels, and extrapolates to those levels that are underground. The RH, U, and V nearest the ground is simply copied downward, while a specified lapse rate is used for temperature and the hypsometric equation is used for geopotential height. The output WPS files will be called `ISOBARIC:YYYY-MM-DDHH` and will contain all the isobaric data (original data above ground, extrapolated data below ground.) A namelist `extrapIsobaric` file is also used as input. *This program is not necessary when processing GEOS-5 model level data, since the GEOS-5 coordinate is terrain following. Users are advised to use the model level data whenever possible.*

- **Splitter utility.**

  - **splitWPS.** A utility that reads in a WPS file and divides the data into new WPS files, which each file containing a single 2D slab of data. The output WPS files will be called `VARNAME.LEVEL:YYYY-MM-DDHH`, where `LEVEL` is the “level code” for the slab. The “level code” follows WPS convention: pressure levels are simply the pressure in Pa; model levels are the indices of the slice (“1” indicates model top in GEOS-5); ground level, 2-meter AGL, and 10-meter AGL are represented as “200100”; and mean sea level is represented as “201300”. A namelist `splitWPS` is also used as input. *This program is not required for preparing data for WPS, but instead allows breaking up a WPS file into individual fields for examination.*

To proceed, the user must first compile the GEOS2WRF software with `

```
./build.sh geos2wrf
```

The user must then review the GEOS-5 data available to them and identify time slices and date/time stamps of interest, and the variables that can be used as-is by WRF. WRF will ultimately require the following fields on either isobaric or GEOS-5 model levels:

- pressure;
- geopotential height;
- horizontal winds;
- temperature; and
- moisture (preferably relative humidity w.r.t. liquid).

Recommended fields that are useful for interpolating or extrapolating near the WRF model terrain level include:

- surface pressure;
- sea level pressure;
5.3 Use of GEOS-5 Meteorological Data

- land-sea mask;
- sea-ice fraction;
- 2-m temperature;
- 2-m relative humidity;
- 10-m horizontal winds;
- skin temperature; and
- terrain height.

With this list in mind, the user must also identify GEOS-5 variables that can be used to derive other variables for WRF. From the utilities listed above, the following derivations can be made:

- Surface geopotential can be used to derive terrain height (via createSOIL-HGT).
- Lake fraction and ocean fraction can be used together to derive a land-sea table (via createLANDSEA).
- Model layer pressure thicknesses can be used to derive model layer pressures (via createPRESSURE).
- Model layer pressure thicknesses can also be used (with model layer temperatures, model layer specific humidity, and the model terrain field) to derive model layer geopotential heights (via createHGT).
- Relative humidity on model levels, isobaric levels, and near ground level can be derived from model, isobaric, and 2-meter temperatures, model, isobaric, and 2-meter specific humidity, and model, isobaric, and surface pressure (via createRH).
- Isobaric temperature, relative humidity, U and V winds can be extrapolated underground (via extrapISOBARIC).

After assembling the list of variables, the user should run GEOS2WPS using a customized namelist.geos2wps for each GEOS-5 file. Execution occurs with a simple ./geos2wps if in the current directory.

After extracting all the GEOS-5 variables, the user must employ the necessary utilities to derive the remaining variables for WRF. The appropriate namelist file (e.g., namelist.createHGT) must be customized, and the user must use the UNIX “cat” command to collect the relevant WPS files together. When ready, the user will execute by typing the program name (e.g., ./createHGT).
### 5.3 Use of GEOS-5 Meteorological Data

The *namelist.geos2wps* file contains the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>&amp;files</strong></td>
<td></td>
</tr>
<tr>
<td>geosFileFormat</td>
<td>Integer, specifies GEOS-5 file format HDF4=1, netCDF3 or netCDF4 = 2, HDFEOS2=4.</td>
</tr>
<tr>
<td>geosFileName</td>
<td>String, specifies GEOS-5 input file name to read</td>
</tr>
<tr>
<td>outputDirectory</td>
<td>String, directory name to write WPS file.</td>
</tr>
<tr>
<td><strong>&amp;coordinates</strong></td>
<td></td>
</tr>
<tr>
<td>longitudeName</td>
<td>String, name of 1-D longitude array in GEOS-5 file.</td>
</tr>
<tr>
<td>latitudeName</td>
<td>String, name of 1-D latitude array in GEOS-5 file.</td>
</tr>
<tr>
<td>hasVerticalDimension</td>
<td>Logical, specifies whether data with vertical dimension are to be processed from GEOS-5 file.</td>
</tr>
<tr>
<td>verticalName</td>
<td>String, name of 1-D vertical coordinate array in GEOS-5 file.</td>
</tr>
<tr>
<td><strong>&amp;forecast</strong></td>
<td></td>
</tr>
<tr>
<td>numberOfTimes</td>
<td>Integer, number of time slices to process from GEOS-5 file.</td>
</tr>
<tr>
<td>validTimes(:)</td>
<td>Array of Strings, specifies valid time(s) of each time slice to process. Format is $YYYY-MM-DD$,$HH$. One array entry should exist for each time slice.</td>
</tr>
<tr>
<td>timeIndices(:)</td>
<td>Array of Integers, specifies time slice indices to process. One array entry should exist for each time slice.</td>
</tr>
<tr>
<td>forecastHours(:)</td>
<td>Array of Integers, specifies nominal forecast hour length for each processed time slice. One array entry should exist for each time slice.</td>
</tr>
<tr>
<td><strong>&amp;variables</strong></td>
<td></td>
</tr>
<tr>
<td>numberOfVariables</td>
<td>Integer, specifies total number of variables to process from the GEOS-5 file.</td>
</tr>
<tr>
<td>variableRanks(:)</td>
<td>Array of Integers, specifies the ranks (number of dimensions) for each GEOS-5 variable to process. Data of rank 3 are assumed to be organized as (lat,lon,time), while rank 4 data are assumed to be organized as (lat,lon,vert,time). One array entry should be assigned for each processed variable.</td>
</tr>
</tbody>
</table>
| variableLevelTypes(:) | Array of Integers, specifies level type for each processed variable. One array entry should be assigned for each variable.  
$=$ 1, ground level  
$=$ 2, 2-meters AGL  
$=$ 3, 10-meters AGL  
$=$ 4, mean sea level  
$=$ 11, model level  
$=$ 12, isobaric level |
5.3 Use of GEOS-5 Meteorological Data

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>variableNamesIn(:)</td>
<td>Array of Strings, specifies name of each processed variable in GEOS-5 file. One array entry should be specified for each variable.</td>
</tr>
<tr>
<td>variableNamesOut(:)</td>
<td>Array of Strings, specifies name of each processed variable as written in the WPS file. One array entry should be specified for each variable. Not that if a processed variable is intended for direct use by WPS (instead of use in deriving something else), the variableNamesOut entry should match that in METGRID.TBL file used by METGRID.</td>
</tr>
<tr>
<td>variableUnits(:)</td>
<td>Array of Strings, specifies units of each processed GEOS-5 variable. One array entry should be specified for each variable. This is included because some GEOS-5 variables are known to be assigned the wrong units when output by the model.</td>
</tr>
<tr>
<td>variableDescriptions(:)</td>
<td>Array of Strings, gives short descriptions of each processed variable as written in the WPS file. One array entry should be specified for each variable.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&amp;subsetData</th>
<th>Logical, specifies whether to process entire GEOS-5 domain or to read and process a subset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>subset</td>
<td>Integer, specifies minimum i (longitude) index of GEOS-5 grid to process. Only used if subset=.true.</td>
</tr>
<tr>
<td>iLonMin</td>
<td>Integer, specifies maximum i (longitude) index of GEOS-5 grid to process. Only used if subset=.true.</td>
</tr>
<tr>
<td>jLatMin</td>
<td>Integer, specifies minimum j (latitude) index of GEOS-5 grid to process. Only used if subset=.true.</td>
</tr>
<tr>
<td>jLatMax</td>
<td>Integer, specifies maximum j (latitude) index of GEOS-5 grid to process. Only used if subset=.true.</td>
</tr>
<tr>
<td>kVertMin</td>
<td>Integer, specifies minimum k (vertical) index of GEOS-5 grid to process. Only used if subset=.true.</td>
</tr>
<tr>
<td>kVertMax</td>
<td>Integer, specifies maximum k (vertical) index of GEOS-5 grid to process. Only used if subset=.true.</td>
</tr>
</tbody>
</table>
The *namelist.temporalInterpolation* file contains the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;all</td>
<td></td>
</tr>
<tr>
<td>fieldName</td>
<td>String, lists name of variable to process.</td>
</tr>
<tr>
<td>&amp;input1</td>
<td></td>
</tr>
<tr>
<td>directory1</td>
<td>String, lists directory with WPS intermediate file.</td>
</tr>
<tr>
<td>prefix1</td>
<td>String, lists prefix of name of WPS intermediate file.</td>
</tr>
<tr>
<td>year1</td>
<td>Integer, lists valid year of WPS intermediate file.</td>
</tr>
<tr>
<td>month1</td>
<td>Integer, lists valid month of WPS intermediate file.</td>
</tr>
<tr>
<td>day1</td>
<td>Integer, lists valid day of WPS intermediate file.</td>
</tr>
<tr>
<td>hour1</td>
<td>Integer, lists valid hour of WPS intermediate file.</td>
</tr>
<tr>
<td>&amp;input2</td>
<td></td>
</tr>
<tr>
<td>directory2</td>
<td>String, lists directory with WPS intermediate file.</td>
</tr>
<tr>
<td>prefix2</td>
<td>String, lists prefix of name of WPS intermediate file.</td>
</tr>
<tr>
<td>year2</td>
<td>Integer, lists valid year of WPS intermediate file.</td>
</tr>
<tr>
<td>month2</td>
<td>Integer, lists valid month of WPS intermediate file.</td>
</tr>
<tr>
<td>day2</td>
<td>Integer, lists valid day of WPS intermediate file.</td>
</tr>
<tr>
<td>hour2</td>
<td>Integer, lists valid hour of WPS intermediate file.</td>
</tr>
<tr>
<td>&amp;output</td>
<td></td>
</tr>
<tr>
<td>directoryOutput</td>
<td>String, lists directory with WPS intermediate file.</td>
</tr>
<tr>
<td>prefixOutput</td>
<td>String, lists prefix of name of WPS intermediate file.</td>
</tr>
<tr>
<td>yearOutput</td>
<td>Integer, lists valid year of WPS intermediate file.</td>
</tr>
<tr>
<td>monthOutput</td>
<td>Integer, lists valid month of WPS intermediate file.</td>
</tr>
<tr>
<td>dayOutput</td>
<td>Integer, lists valid day of WPS intermediate file.</td>
</tr>
<tr>
<td>hourOutput</td>
<td>Integer, lists valid hour of WPS intermediate file.</td>
</tr>
</tbody>
</table>
5.3 Use of GEOS-5 Meteorological Data

The `namelist.createSOILHGT` file contains the following information:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;input</td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
<tr>
<td>surfaceGeopotential-Name</td>
<td>String, name of the surface geopotential field in WPS file.</td>
</tr>
</tbody>
</table>

The `namelist.createHGT` file contains the following information:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;input</td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
<tr>
<td>layerPressure-ThicknessName</td>
<td>String, name of pressure thickness variable between GEOS-5 model levels in the input WPS file.</td>
</tr>
<tr>
<td>layerTemperature-Name</td>
<td>String, name of model layer temperatures in the input WPS file.</td>
</tr>
<tr>
<td>layerSpecific-HumidityName</td>
<td>String, name of model layer specific humidity variable in the input WPS file.</td>
</tr>
<tr>
<td>soilHeightName</td>
<td>String, name of surface terrain variable in the input WPS file.</td>
</tr>
<tr>
<td>modelTopPressure</td>
<td>Real, air pressure (in PA) at very top of GEOS-5 grid. For GEOS-5, this is typically 1 Pa (0.01 mb).</td>
</tr>
</tbody>
</table>
5.3 Use of GEOS-5 Meteorological Data

The `namelist.createLANDSEA` file contains the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;input</code></td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
<tr>
<td>lakeFractionName</td>
<td>String, name of the GEOS-5 variable specifying fraction of grid point covered by lakes in the WPS input file.</td>
</tr>
<tr>
<td>oceanFractionName</td>
<td>String, GEOS-5 variable name specifying fraction of grid point covered by ocean in the WPS input file.</td>
</tr>
</tbody>
</table>

The `namelist.createPRESSURE` file contains the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&amp;input</code></td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
<tr>
<td>layerPressure-ThicknessName</td>
<td>String, names variable with pressure thicknesses between GEOS model levels in the WPS input file.</td>
</tr>
<tr>
<td>modelTopPressure</td>
<td>Real, air pressure (in PA) at very top of GEOS-5 grid. For GEOS-5, this is typically 1 Pa (0.01 mb).</td>
</tr>
</tbody>
</table>
5.3 Use of GEOS-5 Meteorological Data

The `namelist.createRH` file contains the following information:

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;input</td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, lists directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
<tr>
<td>processSurfacePressure</td>
<td>Logical, indicates whether or not to read in surface pressure from the WPS input file.</td>
</tr>
<tr>
<td>onIsobaricLevels</td>
<td>Logical, indicates whether upper air levels are isobaric instead of model level.</td>
</tr>
<tr>
<td>surfacePressureName</td>
<td>String, name of surface pressure variable in WPS input file. Ignored if processSurfacePressure=.false.</td>
</tr>
<tr>
<td>pressureName</td>
<td>String, name of upper-level pressure fields in WPS input file. Ignored if onIsobaricLevels=.true.</td>
</tr>
<tr>
<td>temperatureName</td>
<td>String, name of temperature fields in WPS input file. If 2-meter temperatures are included, then the surface pressure must also be supplied and processSurfacePressure must be set to .true.</td>
</tr>
<tr>
<td>specificHumidityName</td>
<td>String, name of specific humidity fields in WPS input file. If 2-meter specific humidities are included, then the surface pressure must also be supplied and processSurfacePressure must be set to .true.</td>
</tr>
</tbody>
</table>
The *namelist.extrapIsobaric* file contains the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;input</td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, lists directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
<tr>
<td>geopotentialHeightName</td>
<td>String, name of isobaric geopotential height fields in WPS input file.</td>
</tr>
<tr>
<td>temperatureName</td>
<td>String, name of isobaric temperature fields in the WPS file.</td>
</tr>
<tr>
<td>relativeHumidityName</td>
<td>String, name of isobaric relative humidities in the WPS input file.</td>
</tr>
<tr>
<td>uName</td>
<td>String, name of isobaric zonal wind field in WPS input file.</td>
</tr>
<tr>
<td>vName</td>
<td>String, name of isobaric meridional wind field in WPS input file.</td>
</tr>
</tbody>
</table>

The *namelist.splitWPS* file contains the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;input</td>
<td></td>
</tr>
<tr>
<td>directory</td>
<td>String, lists directory for input and output WPS files.</td>
</tr>
<tr>
<td>prefix</td>
<td>String, lists filename prefix of input WPS files.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, lists valid year of WPS file.</td>
</tr>
<tr>
<td>month</td>
<td>Integer, lists valid month of WPS file.</td>
</tr>
<tr>
<td>day</td>
<td>Integer, lists valid day of WPS file.</td>
</tr>
<tr>
<td>hour</td>
<td>Integer, lists valid hour of WPS file.</td>
</tr>
</tbody>
</table>
5.3 Use of GEOS-5 Meteorological Data

A sample script (scripts/discover/run_geos2wrf_merra2_3hrassim.sh) is available to use GEOS2WRF to process 3-hourly MERRA-2 assimilation data. These data are output from GEOS-5 when the global model is adjusting toward a MERRA-2 6-hourly analysis via Incremental Analysis Updates [see section 4.2 of Rienecker et al. (2008)]. This script will run GEOS2WPS, createLANDSEA, createSOILHGT, and createRH to process the data. To run, the user must edit the accompanying config.discover.sh file to set the path to the NU-WRF code, the work directory, and the modules used to compile GEOS2WRF; then, the run_geos2wrf_merra2_3hrassim.sh should be modified to specify the start and end dates and hours to process. (Users who wish to use 6-hourly MERRA or MERRA-2 data can use either GEOS2WRF or MERRA2WRF; however, the 3-hourly MERRA-2 data can only be processed with GEOS2WRF.)

5.3.2 MERRA2WRF

MERRA2WRF is a monolithic program customized to process the 6-hourly reanalyses from MERRA and MERRA-2. It was first developed for MERRA under the assumption that the archived data files (called “collections” in GEOS-5 terminology) would be permanent, making it possible to build a single robust preprocessing tool. Recently support was added for 6-hourly MERRA-2 fields; however, 3-hourly MERRA-2 processing is not possible due to significant differences in the data collections (users must fall back to GEOS2WRF for these 3-hourly data).

MERRA and MERRA-2 files are accessible from the NASA GES DISC web page (http://disc.sci.gsfc.nasa.gov/daac-bin/DataHoldings.pl), and are
5.3 Use of GEOS-5 Meteorological Data

available to the general public. MERRA-2 files are also accessible on the NASA Discover supercomputer in /discover/nobackup/projects/gmao/merra2/merra2/scratch/, but are only available to select users authorized by the GMAO.

MERRA2WRF must be compiled using ./build.sh merra2wrf. The following files must then be gathered from the MERRA or MERRA-2 datasets:

- **const2d_asm_Nx** (in HDFEOS2 or NETCDF format):
  - 'XDim' or 'lon' (longitude)
  - 'YDim' or 'lat' (latitude)
  - 'PHIS' (surface geopotential)
  - 'FRLAKE' (lake fraction)
  - 'FROCEAN' (ocean fraction)

- **inst63d_ana_Nv** (variable names are HDF4 or netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hour)
  - 'levels' or 'Height' or 'lev' (nominal pressure for each model level)
  - 'ps' or 'PS' (surface pressure)
  - 'delp' or 'DELP' (layer pressure thicknesses)
  - 't' or 'T' (layer temperature)
  - 'u' or 'U' (layer eastward wind)
  - 'v' or 'V' (layer northward wind)
  - 'qv' or 'QV' (layer specific humidities)

- **inst63d_ana_Np** (variable names are HDF4 or netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hours)
  - 'slp' or 'SLP' (sea level pressure)

- **tavg12d_slv_Nx** (variable names are HDF4 or netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hours)
  - 'u10m' or 'U10M' (10-meter eastward wind)
  - 'v10m' or 'V10M' (10-meter northward wind)
  - 't2m' or 'T2M' (2-meter temperature)
5.3 Use of GEOS-5 Meteorological Data

- 'qv2m' or 'QV2M' (2-meter specific humidity)
- 'ts' or 'TS' (skin temperature)

• `tavg1_2d_ocn_Nx` (variable names are HDF4/netCDF or HDFEOS2):
  - 'longitude' or 'XDim' or 'lon' (longitude)
  - 'latitude' or 'YDim' or 'lat' (latitude)
  - 'time' or 'TIME:EOSGRID' or 'TIME' (synoptic hours)
  - 'frseaice' or 'FRSEAICE' (sea ice fraction)

Note that the `tavg1_2d_slv_Nx` and `tavg1_2d_ocn_Nx` collections are 1-hour averages that are valid at the bottom of the hour. For simplicity, MERRA2WRF uses the 00:30Z average data with the 00Z instantaneous fields, the 06:30Z average data with the 06Z instantaneous fields, and so on.

The user can run `utils/geos2wrf2/RUN_MERRA/Run_MERRA.csh` to ftp the MERRA data and run MERRA2WRF from a specified start date and end date using the command `./Run_MERRA.csh StartDate EndDate OutputDir NUWRFDIR`. A namelist file will be created for each processing date, and files readable for METGRID will be generated.

To use MERRA-2 reanalysis, the user may run `utils/geos2wrf2/RUN_MERRA2/proc_merra2_ges_disc.sh` to download the MERRA-2 files from the GES DISC web page and process them with MERRA2WRF. The command is `./proc_merra2_ges_disc.sh StartDate EndDate RunDir NUWRFDIR`. Alternatively, the user can run `utils/geos2wrf2/RUN_MERRA2/Run_MERRA2.csh` on Discover to copy the MERRA2 files from the GMAO MERRA2 Discover directory and run MERRA2WRF. The command is `Run_MERRA2.csh StartDate EndDate OutputDir NUWRFDIR`. A namelist file will be created for each processing date, and files readable for METGRID will be generated.

A third alternative is for the user to customize the `utils/geos2wrf2/namelist/namelist.merra2wrf` by hand to process the selected MERRA files and `namelist.merra2_2wrf` to process the selected MERRA-2 files. The namelist files consist of a single block.
5.3 Use of GEOS-5 Meteorological Data

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outputDirectory</td>
<td>String, lists directory for writing WPS output files.</td>
</tr>
<tr>
<td>merraDirectory</td>
<td>String, lists directory containing MERRA or MERRA-2 input files.</td>
</tr>
<tr>
<td>merraFormat_const_2d_asm_Nx</td>
<td>Integer, specifies format of const_2d_asm_Nx file. 1=HDF4, 2=netCDF, 4=HDFEOS2.</td>
</tr>
<tr>
<td>merraFile_const_2d_asm_Nx</td>
<td>String, name of const_2d_asm_Nx file.</td>
</tr>
<tr>
<td>numberOfDays</td>
<td>Integer, lists number of days to process. Each MERRA or MERRA-2 collection. (excluding const_2d_asm_Nx) will have one file per day.</td>
</tr>
<tr>
<td>merraDates(:)</td>
<td>Array of strings, list each day to be processed (format is YYYY-MM-DD).</td>
</tr>
<tr>
<td>merraFormat_inst6_3d_ana_Nv</td>
<td>Integer, specifies format of inst6_3d_ana_Nv files. 1=HDF4, 2=netCDF, 4=HDFEOS2.</td>
</tr>
<tr>
<td>merraFiles_inst6_3d_ana_Nv(:)</td>
<td>Array of strings, specifying names of inst6_3d_ana_Nv files.</td>
</tr>
<tr>
<td>merraFormat_inst6_3d_ana_Np</td>
<td>Integer, specifies format of inst6_3d_ana_Np files. 1=HDF4, 2=netCDF, 4=HDFEOS2.</td>
</tr>
<tr>
<td>merraFiles_inst6_3d_ana_Np(:)</td>
<td>Array of strings, specifying names of inst6_3d_ana_Np files.</td>
</tr>
<tr>
<td>merraFormat_tavg1_2d_slv_Nx</td>
<td>Integer, specifies format of tavg1_2d_slv_Nx files. 1=HDF4, 2=netCDF, 4=HDFEOS2.</td>
</tr>
<tr>
<td>merraFiles_tavg1_2d_slv_Nx(:)</td>
<td>Array of strings, specifying names of tavg1_2d_slv_Nx files.</td>
</tr>
<tr>
<td>merraFormat_tavg1_2d_ocn_Nx</td>
<td>Integer, specifies format of tavg1_2d_ocn_Nx files. 1=HDF4, 2=netCDF, 4=HDFEOS2.</td>
</tr>
<tr>
<td>merraFiles_tavg1_2d_ocn_Nx(:)</td>
<td>Array of strings, specifying names of tavg1_2d_ocn_Nx files.</td>
</tr>
</tbody>
</table>

The software is run by typing ./merra2wrf namelist.merra2wrf. The output files will be named MERRA:$YYYY-$MM-$DD-$HH, where $YYYY is the four-digit year, $MM is the two-digit month, $DD is the two-digit day, and $HH is the two-digit hour. These files are readable by METGRID.
5.4 Use of New Erodible Soil Options

NU-WRF includes several new options for specifying erodible soil (EROD) for dust emissions (Kim et al., 2014). The workflow depends a bit on the particular option selected, but requires compilation of WRF-Chem and WPS (.build.sh chem wps if using normal chemistry, or .build.sh kpp wps if using KPP chemistry).

The four available EROD options are:

- **EROD_STATIC**. This is the EROD option inherited from the community WRF. Annual EROD at 0.25 deg resolution for sand, silt, and clay is processed and fed to WRF-Chem.

- **EROD_MDB**. This is a new seasonal EROD dataset derived from MODIS-Deep Blue climatological aerosol products. [See Ginoux et al. (2012) and Ginoux et al. (2010) for description of estimating frequency of occurrence of optical depth – these are converted to EROD]. Data are subdivided into three groups (for sand, silt, and clay) at 0.1 deg resolution for four meteorological seasons (December-January-February, March-April-May, June-July-August, September-October-November). These are processed and passed to WRF-Chem.

- **EROD_DYN_CLIMO**. This is a new “dynamic climatological” EROD option. It uses a monthly surface bareness field derived at 30 arc second resolution from the community WRF climatological MODIS vegetation fraction dataset (greenness_fpar_modis/), with adjustments from the
5.4 Use of New Erodible Soil Options

community WRF’s soil type and MODIS and USGS land use datasets to screen out water bodies. It also uses a 30 arc second topographic depression dataset derived from the community WRF’s terrain dataset. These fields are passed to WRF-Chem, which will create an instantaneous EROD field from these variables with adjustments to screen out snowy or very cold locations.

- **EROD_DYN**. This is a new “dynamical” EROD option. It uses the same topographic depression field as **EROD_DYN_CLIMO**, plus a surface bareness field based on either NASA GIMMS or NASA SPoRT daily NDVI products. The data are passed to WRF-Chem, which will construct an instantaneous EROD field from the bareness and topographic depression with adjustments to screen out snowy or very cold locations.

A sample workflow for EROD is presented here:

- **Assemble GEOG data.** Several new EROD-related fields must be obtained from the NU-WRF group and placed in subdirectories with the standard GEOG data available with the community WRF. For **EROD_MDB**, they are `erod_mdb_clay_0.1deg/`, `erod_mdb_sand_0.1deg/`, and `erod_mdb_silt_0.1deg/`. For **EROD_DYN_CLIMO**, they are `bareness_dyn_climo/` and `TOPODEP_30s/`. For **EROD_DYN**, the `TOPODEP_30s/` data must be staged; in addition, the user is advised to collect NDVI-based greenness in GEOGRID format from NASA SPoRT and place the files in a `gvfsport/` directory.

- **Create NDVI-based bareness for EROD_DYN option.** The NDVIBARENESS4WRF tool is used to process NDVI data and generate a bareness field. Both bareness and NDVI are written in WPS intermediate format for direct ingest into METGRID. The NDVIBARENESS4WRF code is described by Kemp (2016).

- **GEOGRID.** Run for terrestrial processing. Use the `GEOGRID.TBL.ARW_CHEM_NUWRF` file to ensure EROD related fields are specified. If preparing for the **EROD_DYN** option, “gvfsport” should be specified as the first part of the “geog data res” option in `namelist.wps` – this will ensure the SPoRT greenness is processed rather than the climatological greenness data available with the community WRF.

- **UNGRID.** Run for normal GRIB file processing.

- **METGRID.** Run for normal processing. If preparing for the **EROD_DYN** option, user should modify `namelist.wps` to include the file prefix(es) for the bareness data in the “fg name” namelist option. Note that the `METGRID.TBL.ARW` file in NU-WRF has been modified to recognize the new EROD-related fields.

- **REAL.** Run to produce initial and lateral boundary conditions. Namelist variable “chem_opt” should be set to 401 for simple dust treatment. (Dust emissions can also be used with GOCART chemistry by setting “chem_opt”
5.5 Use of GOCART Aerosol Data

NU-WRF offers advanced aerosol modeling using the implementation of GOCART [see Chin et al. (2002) and Ginoux et al. (2001)] in WRF-Chem. Running GOCART in WRF allows for aerosol coupling with the Goddard 3ICE and 4ICE microphysics schemes and with the 2011 and 2014 Goddard radiation schemes, providing simulation of the direct and indirect aerosol effects on weather and climate. For best results, it is necessary to provide initial and lateral boundary conditions for GOCART, plus surface based emissions. To that end, the NU-WRF modeling system includes the new GOCART2WRF preprocessor for providing chemical boundary conditions from the GEOS-5, and includes the community PREP_CHEM_SOURCES program for emissions. To run, the user must compile with 

```
./build.sh chem wps gocart2wrf prep_chem_sources plot_chem
```

GOCART2WRF supports four possible sources of GOCART data:

- offline GOCART;
- on-line GOCART from GEOS-5;
- MERRAero reanalyses [Kishcha et al., 2014]; and

GOCART aerosol data can be used to simulate the effects of aerosols on weather and climate. The new namelist variable “erod_option” in the &chem block should be set to “static”, “mdb”, “dyn_climo”, or “dyn”. If not set, “static” is assumed.

- **WRF.** Run for EROD simulation. Optionally set the new “gocart_dustemiss_layer” namelist variable to highest vertical model level to spread dynamic dust (default is 1), or set “gocart_dustemiss_suppress=1” to shut off dynamic dust emissions. The new variable EROD_TIMESTEP in the wrfout netCDF file will show the instantaneous EROD field for sand, silt, and clay for whatever EROD option is used. Other variables of note are:
  - BARENESS_DYN_CLIMO: Monthly input bareness field for EROD_DYN_CLIMO option.
  - EROD_MDB_CLAY: Seasonal EROD for clay for EROD_MDB option.
  - EROD_MDB_SAND: Seasonal EROD for sand for EROD_MDB option.
  - EROD_MDB_SILT: Seasonal EROD for silt for EROD_MDB option.
  - EROD_STATIC: The standard EROD field available with the community WRF.
  - TOPODEP: The topographic depression values for sand, silt, and clay for the EROD_DYN_CLIMO and EROD_DYN options.
5.5 Use of GOCART Aerosol Data


A workflow supporting the use of GOCART in NU-WRF might look like this:

- **WPS.** Perform terrestrial and meteorological preprocessing as normal.

- **REAL.** Generate meteorological initial and lateral boundary conditions as normal.

- **GOCART2WRF.** Process GOCART data and insert into output files from REAL. Typically the user must edit a `namelist.gocart2wrf` to specify the number of WRF domains, location of REAL output files, and location, source, and file prefixes of the GOCART files. GOCART2WRF is then executed at the command line using `./gocart2wrf` with `namelist.gocart2wrf` in the current working directory. GOCART2WRF will obtain the required dates and times from the REAL netCDF files, search the GOCART files for the corresponding dates and times, read and interpolate the required GOCART variables, and essentially append those fields to the REAL files. Currently 17 GOCART variables are processed: Hydrophobic and Hydrophilic Black Carbon, Hydrophobic and Hydrophilic Organic Carbon, dust particles with 0.5 µm, 1.4 µm, 2.4 µm, 4.5 µm, and 8.0 µm effective radii, sea salt particles with 0.3 µm, 1.0 µm, 3.2 µm, and 7.5 µm effective radii, and concentrations of Dimethyl Sulfide, Methanesulfonic Acid, Sulfur Dioxide, and Sulfate.

The `namelist.gocart2wrf` file contains the following entries:
5.5 Use of GOCART Aerosol Data

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;wrf</td>
<td>Integer, specifies number of WRF domains.</td>
</tr>
<tr>
<td>wrf_dir</td>
<td>String, specifies directory with wrfinput and wrfbdy files.</td>
</tr>
<tr>
<td>&amp;gocart_shared</td>
<td>Integer, specifies format of GOCART data; currently must be set to 5 (for netCDF4 files)</td>
</tr>
<tr>
<td>gocart_format</td>
<td>String, lists source of GOCART data. 4 options are available: 'GEOS' : Output from GEOS-5 (default) 'MERRA2' : Output from MERRA-2 reanalysis 'MERRAERO' : Output from MERRAero reanalysis 'OFFLINE' : Output from offline GOCART</td>
</tr>
<tr>
<td>gocart_dir</td>
<td>String, specifies directory with netCDF4 GOCART files to process. If processing MERRAero or MERRA2 data, files are assumed to be in subdirectories, e.g., For MERRAERO: gocart_dir/Y2005/M05/ For MERRA-2: gocart_dir/$STREAM/stage/Y2005/M05/ where $STREAM is MERRA2_100 for data before 1992, MERRA2_200 for data from 1992 to 2000, MERRA3_300 for data from 2001 to 2010, or MERRA2_400 for data from 2011 on.</td>
</tr>
<tr>
<td>gocart_prefix</td>
<td>String, specifies file name prefix for netCDF4 GOCART files. Ignored if processing MERRA-2 or MERRAero data.</td>
</tr>
</tbody>
</table>

If processing MERRA-2 data, the user can run `utils/gocart2wrf_2/scripts/proc_merra2_gocart_ges_disc.sh` to download the MERRA-2 GOCART files from the GES DISC web page, and process them with GOCART2WRF. The command is `/proc_merra2_gocart_ges_disc.sh NumDomains StartDate EndDate RunDir NUWRFDIR`. A namelist file will be created by the script, and updated wrfbdy and wrfinput netCDF files will be created.

- **PREP_CHEM_SOURCES.** This community tool processes a number of biogenic, anthropogenic, volcanic, and wildfire emissions (Freitas et al., 2011). Operating this program is largely described in Peckham et al. (2015b) and Peckham et al. (2015a), and requires customizing a `prep_chem_sources.inp` file and downloading emissions data supported by the program. The NU-WRF version has several modifications:

  - **Map projection.** The map projection code from WPS has been added to PREP_CHEM_SOURCES to ensure consistency in emission interpolation. This support is automatic when using the NU-WRF build system, and no user action is required.
  
  - **Improved GOCART background fields.** Processing emissions
5.5 Use of GOCART Aerosol Data

for GOCART requires monthly climatological background fields of Hydrogen Peroxide, Hydroxide, and Nitrate. The community data for PREP_CHEM_SOURCES contains 55 vertical levels, is only for 2006, and is missing some information useful for vertical interpolation. New 72-level datasets produced by GSFC are now available in two formats with improved vertical interpolation information. The user should edit `prep_chem_sources.inp` and set the new “gocart_bg_data_type” variable to “old” or “new” to switch between the community and new files. If “new” files are selected, they must be in a `gocart_bg/new/` directory on the same level as the `gocart_bg/` folder containing the community 55-level files. If “new” is selected, the program will first search for `geos5_met.1MAG_YYYYYMM.nc` and `gmi_merra_oxidants_YYYYMM_1.25x1.nc` files, where YYYY is the 4-digit year and MM is the 2-digit month. If the program fails to find both files, it will fall back on `gmi_2006MM.nc` files, which are also new 72-level files but only exist for 2006.

– Improved interpolation of GOCART background fields. The community PREP_CHEM_SOURCES program uses simple averaging of GOCART background grid points to each WRF grid point, which implicitly assumes the WRF grid is at a coarser resolution than the background. This results in unphysical blocky fields, which are further marred by gradients associated with sharp WRF terrain. In the NU-WRF version of PREP_CHEM_SOURCES, bilinear interpolation is used in cases where less than two background grid points are averaged to a WRF grid point, leading to much smoother fields.

– GFEDv3.1. The GFEDv3.1 biomass burning emissions dataset [van der Werf et al. (2010) and Randerson et al. (2013)] is supported. The user must edit the `prep_chem_sources.inp` file, toggle the new variable “use_gfedv3=1”, set the new “gfedv3_data_dir” variable to specify the directory with the GFEDv3.1 data, and edit the new “gfedv3_suffix” variable to list the species to process (e.g., “BC,OC,PM2p5,SO2”). Note that the species are used to construct the names of the monthly emissions files (e.g., `GFED3.1_200301_BC.txt`). Also, note that GFEDv3.1 and GFEDv2 data cannot be used simultaneously. Currently the following species can be processed: BC, C2H4, C2H4O, C2H4, C2H5OH, C2H6S, C2H6, C3H6O, C3H6, C3H8, CH4, C5H8, CH2O, CH3OCH, CH4, CO2, CO, NH3, NOx, OC, PM2p5, SO2, Terpenes, Toluene, and TPM. DM can also be processed but is not currently used by WRF-Chem.

– QFED. The NASA QFED wildfire emissions dataset [Darmenov and da Silva (2013)] is supported. The user must edit the `prep_chem_sources.inp` file, toggle the new variable “use_qfed=1”, set the new “qfed_data_dir” variable to specify the top directory with QFED data (yearly subdirectories like `Y2005/` are assumed, with monthly subdirectories like `M05/` within each annual subdirectory), and edit the new “qfed_suffix”
5.5 Use of GOCART Aerosol Data

variable to list the species to process (e.g., “bc,oc,pm25,so2”). Note that the species are used to construct the names of the emissions files (e.g., `qfed2.emis_bc.005.20140404.nc4`), so the names are case sensitive. Also, note that QFED cannot be used simultaneously with GFEDv3.1 or GFEDv2. Currently the following QFED species can be processed: acet, ald2, alk4, bc, c2h6, c3h6, c3h8, ch2o, ch4, co, co2, mek, nh3, no, oc, pm25, and so2.

- **Output of map projection data.** New `.map` files with map projection data are automatically output. These files are intended for use by PLOT_CHEM to visualize the fields.

- **PLOT_CHEM.** This is an optional step to create simple visualizations of emissions output from PREP_CHEM_SOURCES. The program reads in a GrADS control file produced by PREP_CHEM_SOURCES, the corresponding GrADS binary file, and the special `.map` file with critical map projection information. PLOT_CHEM will then create visualizations of each field using NCAR Graphics. The plots are not publication quality and are only intended for sanity checking. To run, the user must first create a symbolic link `grads.ctl` to the desired GrADS control file, and then run `./plot_chem` in the same directory as the GrADS and `.map` files. The output is a `gmeta` file which can be viewed using the NCAR Graphics `idt` program (see [http://ngwww.ucar.edu](http://ngwww.ucar.edu) for information on NCAR Graphics).

- **CONVERT_EMISS.** This is a community WRF-Chem preprocessor that takes the output from PREP_CHEM_SOURCES and rewrites the fields in new netCDF files for reading by WRF-Chem. This program is described in [Peckham et al.] and [Peckham et al.], and requires modifying a `namelist.input` file to specify domain information, physics, and chemistry options. The program is then run in the same directory as the `namelist.input` and the PREP_CHEM_SOURCES output files.

There are two issues to keep in mind:

- First, CONVERT_EMISS does not currently process more than one domain at a time. Thus, the user must process each domain in separate executions, and must rename the input files to use “d01” before execution, regardless of what the actual domain number is. The output files must then be renamed to restore the actual domain numbers for WRF-Chem. In addition, a `namelist.input` file must be customized for each execution with the “max_dom” variable set to “1” and the first grid domain variables (e.g., “dx”, “nx”, etc) conforming to whichever domain is being processed.

- Second, the output files from PREP_CHEM_SOURCES must be renamed to conform to the naming convention expected by CON-
5.5 Use of GOCART Aerosol Data

VERT_EMIT. The naming convention for the different emissions files are documented in [Peckham et al.](2015b).

The NU-WRF version of CONVERT_EMIT includes support for most GFEDv3.1 and QFED species without any special action required by the user. However, if the user processes C2H4, CH4, CO2, terpenes, or DM emissions from GFEDv3.1 with PREP_CHEM_SOURCES – or ch4 or co2 from QFED – then the new namelist.input variable “gfedv3_biomass_burn_extravars” in the &chem block must be set to 1. This will cause CONVERT_EMIT to read the additional fields at the end of the emissfire binary file (these are not produced by the community version of PREP_CHEM_SOURCES, so reads are not attempted by default to preserve backward compatibility).

In addition, if PREP_CHEM_SOURCES processed “new” GOCART background files, the &chem block in namelist.input must be modified to add “new_gocart_bg_files=.true.” so CONVERT_EMIT will read 72 levels of data and perform correct vertical interpolation. If this setting is omitted, the background fields output by CONVERT_EMIT will likely contain unphysical values.

- **WRF-Chem.** Running WRF-Chem is similar to the basic case in section 5.1 but requires the &chem namelist block to be included in the namelist.input with with GOCART activated (chem_opt = 300, 301, 302, or 303). Aerosol coupling with the Goddard 3ICE or 4ICE schemes (mp_physics=55 or 56) will be activated if the new “gsfcgee_gocart_coupling” namelist variable is set to 1 (set by default). Likewise, aerosol coupling with the 2011 or 2014 Goddard radiation schemes (ra_lw_physics=55 or 56 and ra_sw_physics=55 or 56) will be activated if the new “gsfcrad_gocart_coupling” namelist variable is set to 1 (set by default).
5.6 Use of CASA CO₂ Data

NU-WRF has a new capability for running simulations with CO₂ treated as a tracer (i.e., no interaction with physics). This requires specifying initial, lateral boundary, and flux emission fields of CO₂. To that end, several utilities have been written to process CASA global climatological CO₂ concentrations and flux emissions and provide them to WRF-Chem: READ_CO2_CONC, READ_CO2_FLUX, and CASA2WRF.

A second pre-processing capability is added to the CASA2WRF utility in which the temporal interpolation of CASACO2 flux is depending on the NU-WRF model state. This option is initiated by a namelist option “flux_interpolate”. With this option, the NU-WRF model state is read by CASA2WRF every hour and wrf-domain interpolated CO₂ flux components from 6 different sources - Respiratory (monthly), Net Production (monthly), Bio-fuel (monthly), Fossil Fuel (yearly), Wild fire (daily) and Ocean CO₂ are combined to produce a total flux and flux tendency netCDF files to input to NU-WRF WRF-Chem runs.

To compile, the user must type `./build.sh casa2wrf`. It creates the following executables in `utils/casa2wrf/`:

- `pproc/Read_CO2_conc.x`: To pre-process the CO₂ concentration data file to netCDF data file.
- `pproc/Read_CO2_Flux.x`: To pre-process the CO₂ flux data file to netCDF data files.
5.6 Use of CASA CO$_2$ Data

- *pproc1/ConvertData2Netcdf.x*: To pre-process the CO$_2$ flux component files in binary format to netCDF data files.

- *bin/casa2wrf*: To process the CO$_2$ concentration and flux data to *wrfinput*, *wrfbdy*, and flux input files for WRF-Chem runs.

Instructions for running these programs, including namelist definitions, are provided in [Tao et al. (2014)](#). A sample workflow is provided below:

- **WPS.** Perform terrestrial and meteorological preprocessing as normal.

- **REAL.** Generate meteorological initial and lateral boundary conditions as normal.

- **READ_CO2_CONC.** Reads the CASA CO$_2$ concentration files in flat binary format, and converts to netCDF with a time stamp.

- **READ_CO2_FLUX.** Reads the CASA CO$_2$ flux files in flat binary format, and converts to netCDF with a time stamp.

- **ConvertData2Netcdf.** Reads the CASA CO$_2$ flux files from different sources in flat binary format, and converts to netCDF with a time stamp.

- **CASA2WRF.** Read the CO$_2$ netCDF files, interpolates concentration and flux data to the WRF grids (single or nested), read the *wrfout* files from a NU-WRF run every hour or any specified time interval from namelist, temporally interpolate the NPP and RESP CO$_2$ fluxes with relation the NU-WRF state, combine the fluxes from all sources, calculates the rates of change of flux/hour at the user specified time frequency, appends the interpolated concentrations to the initial and lateral boundary condition netCDF files (*wrfinput* and *wrfbdy*), and writes the interpolated fluxes to a new netCDF file (*CO2_domain_date*).

- **WRF-Chem.** Run WRF-Chem with CASA CO$_2$ chemistry options in the *namelist.input* (including chem_opt = 18, emiss_opt=18, and emiss_inpt_opt=18). Other appropriate settings are listed in [Tao et al. (2014)](#).

![Casa2wrf – wrfchem workflow](image)

---

60
5.7 Use of SPoRT Sea Surface Temperature Data

NASA SPoRT generates sea surface temperature analyses for the northern western hemisphere every 12 hours valid at 06Z and 18Z. A script `fetch_sport_sst_northwestHemi.py` is available to download GRIB2 files of these analyses from the SPoRT FTP site. The script and companion configuration file `sport_sst_northwestHemi.cfg` are in the `scripts/fetch_data/` directory. The configuration file specifies the start and end dates and hours for downloading data, along with the top directory for storing the downloaded files. The user can override the date/time information in this file by running `fetch_sport_sst_northwestHemi.py` with a `-L` flag (or equivalently, `--latest`) on the command line; this will cause the script to fetch all available files valid for the most recent 24 hours. Files will be uncompressed via gunzip after being downloaded.

The uncompressed GRIB2 files can be processed by UNGRIB (we recommend using the prefix “SPORT_SST” for the output files). The user must then edit the `namelist.wps` file to include the SPoRT SST prefix in the “fg_name” namelist variable so METGRID processes the data along with other atmospheric fields. When run, METGRID will replace the SST or skin temperature from atmospheric fields with that from the SPoRT analysis. The remaining steps (REAL and WRF) can be completed as normal.

5.8 Use of RSS Sea Surface Temperature Data

A special preprocessor included with NU-WRF is SST2WRF, which processes several sea surface temperature (SST) products produced by Remote Sensing Systems (RSS; see [http://www.remss.com](http://www.remss.com)). These products are potential alternatives to the SST or skin temperature fields often provided in meteorological GRIB files (e.g., from the NOAA GFS or NAM models). Because RSS products are not available in GRIB format, UNGRIB cannot process them and a tool like SST2WRF is required as a substitute.

SST2WRF currently supports several different analysis products classified by source instrument and by algorithm version. The instrument SST analyses are

- **mw_ir.** 9-km global SST valid at 1200 UTC based on microwave (TMI, AMSR-E, AMSR2, WindSat) and Infrared (Terra MODIS, Aqua MODIS) data.

- **mw.** 25-km global SST valid at 0800 LT, based on Microwave (TMI, AMSR-E, AMSR2, WindSat) data.

The algorithm versions are:

- **rt.** The real-time algorithm.

- **v04.0.** Version 4 algorithm.

A workflow for SST2WRF could be similar to that in section 5.1, but would require running SST2WRF in addition to UNGRIB. UNGRIB is responsible for processing meteorological fields, while SST2WRF will process only the SST related fields from RSS. (One could also replace UNGRIB with GEOS2WRF or MERRA2WRF; for simplicity, we will assume UNGRIB is used.)

The user must compile using `.build.sh sst2wrf`. A script given in `scripts/discover/fetch_rss_sst.sh` can be used to download SST data by typing `fetch_rss_sst.sh startdate enddate instrument type`. The retrieved files can then be processed using the batch script at `scripts/discover/run_sst2wrf.discover.sh`. As an alternative, the user can customize a sample namelist file given in `utils/sst2wrf/namelist/namelist.sst2wrf` to provide the following information:

<table>
<thead>
<tr>
<th>Variable Names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;input</td>
<td></td>
</tr>
<tr>
<td>instrument</td>
<td>String, specifies instrument(s) used for analysis; options are “mw_ir”, “mw”.</td>
</tr>
<tr>
<td>year</td>
<td>Integer, specifies valid year of analysis.</td>
</tr>
<tr>
<td>dayOfYear</td>
<td>Integer, specifies valid day of year of analysis.</td>
</tr>
<tr>
<td>version</td>
<td>String, specifies algorithm for analysis; options are “rt”, “v04.0”.</td>
</tr>
<tr>
<td>inputDirectory</td>
<td>String, specifies directory with SST file.</td>
</tr>
<tr>
<td>&amp;output</td>
<td></td>
</tr>
<tr>
<td>outputDirectory</td>
<td>String, specifies directory for output file.</td>
</tr>
<tr>
<td>prefixWPS</td>
<td>String, specifies file name prefix for output; prefix must also be in <code>METGRID.TBL</code> for METGRID to process.</td>
</tr>
<tr>
<td>&amp;fakeoutput</td>
<td></td>
</tr>
<tr>
<td>numFakeHours</td>
<td>Integer, specifies number of hours of each day that additional WPS files should be written for. Currently only one SST analysis is available per day, but METGRID requires all time varying fields to have same time interval. Thus, we optionally output daily SST at multiple times corresponding to when atmospheric data are available.</td>
</tr>
<tr>
<td>fakeHours</td>
<td>Array of integers, specifies nominal hours of day in UTC for an input daily SST analysis to be output.</td>
</tr>
</tbody>
</table>

The program is then run by typing `.sst2wrf` in the same directory as `namelist.sst2wrf`.

The resulting output files will be in WPS intermediate format. The user must then edit `namelist.wps` and list both the meteorological (from UNGRIB) and SST (from SST2WRF) file prefixes in the “fg_name” namelist variable [see Chapter 3 of NCAR (2016)]. METGRID will replace the SST or skin temperature from UNGRIB with that from SST2WRF. The remaining steps (REAL and WRF) can be completed as normal.
5.9 Initializing Small Inland Lake Temperatures

Proper initialization of lake temperatures can be challenging, as small water bodies may not be resolved by external datasets (GRIB, MERRA2, RSS SST, etc). By default, METGRID will extrapolate skin temperature from externally resolved water points; but this can lead to anomalously warm or cold lake temperatures, particularly far inland from large water bodies. To avoid this problem, WPS provides another approach [from the “Alternative Initialization of Lake SSTs” subsection in Chapter 3 of NCAR (2016)]:

- Special landuse datasets (usgs_lakes and modis_lakes) that discriminate between lakes and oceans. We recommend using the “modis_lakes” dataset when running GEOGRID.

- A special preprocessor called AVG_TSFC, which reads multiple WPS intermediate files (from UNGRIB, MERRA2WRF, etc) and calculates daily-average air temperature (TAVGSFC). TAVGSFC can then be passed with other data to METGRID for horizontal interpolation. REAL can then use TAVGSFC for the surface temperatures over lakes. While there is room for experimentation, feeding data from the previous 4 weeks to AVG_TSFC should provide reasonable estimates for lake temperatures.

The attached figure illustrates use of AVG_TSFC within a simple WRF workflow. This can be combined with other workflows, including use of SST2WRF.
5.9 Initializing Small Inland Lake Temperatures

In this latter case, the SST field will be used for oceans and large lakes (e.g., the Great Lakes), while the TAVGSFC data will be used for small lake temperatures.

Simple AVG_TSFC workflow
(can also combine with WRF-LIS, MERRA, GEOS, CASA, SST2WRF and/or GOCART workflow)
6 Post-Processors

In Section 5, sample workflows are presented to initialize and run WRF in multiple configurations. In the present section we address the question of post-processing the WRF (and in some cases, LIS) output. All of the post-processors address the task of evaluation, either subjective or objective. Several tools are available to prepare visualizations of model fields, while others allow for calculating verification metrics against observations or gridded analyses.

It should be noted that other generic tools exist that can be used to evaluate NU-WRF netCDF4 output. These include:

- IDL (http://www.harrisgeospatial.com/ProductsandSolutions/GeospatialProducts/IDL.aspx);
- IDV (http://www.unidata.ucar.edu/software/idv/);
- MATLAB (http://www.mathworks.com/products/matlab/index.html?ep_tid=gn_loc_drop);
- NCL (http://www.ncl.ucar.edu);
- Ncview (http://meteora.ucsd.edu/~pierce/ncview_home_page.html);
- Python (http://www.python.org) with:
  - Matplotlib (http://matplotlib.org) or;
  - PyNGL and PyNIO (https://www.pyngl.ucar.edu);
- R (http://www.r-project.org) with:
  - ncdf4 (https://cran.r-project.org/web/packages/ncdf4/index.html); or
  - RNetCDF (https://cran.r-project.org/web/packages/RNetCDF/index.html);
- VAPOR (https://www.vapor.ucar.edu).

See also a list of netCDF compatible software maintained at http://www.unidata.ucar.edu/software/netcdf/software.html.

6.1 G-SDSU

The Goddard Satellite Data Simulator Unit [G-SDSU; Matsui et al. (2014)] is a program developed by NASA for use with cloud-resolving model data. The program can simulate multiple microwave, radar, visible and infrared, lidar, and broadband satellite products from the input model fields. These simulations can be used for detailed verification against actual satellite observations [Matsui et al., 2009], for assimilation of satellite radiances, or for exploring future satellite missions. The software is compiled by typing ./build.sh gsdsu. Instructions on running the program are available in Matsui and Kemp (2016).
6.2 RIP4

The community Read/Interpolate/Plot Version 4 software package is capable of processing WRF netCDF files, deriving new variables (e.g., air temperature, relative humidity, CAPE), interpolating to isobaric, isentropic, or constant height levels as well as vertical cross-sections, and plotting the fields in NCAR Graphics gmeta format. Advanced options also exist, including calculating and plotting trajectories, interpolating between coarse and fine grid resolutions, and writing data in a format readable by the Vis5D visualization package (see http://vis5d.sourceforge.net).

The RIP software is compiled by typing ./build.sh rip. The two most important executables in RIP4 are:

- **RIPDP_WRFARW**: This program will read WRF netCDF files and transform the data to an internal binary data format. The user will have the option of processing either a basic set of variables or all the variables in the files.
- **RIP**: This program will process the output of RIPDP_WRFARW based on the user’s settings in a provided input file.

Users are referred to Stoelinga (2006) for detailed instructions on using RIP. Sample namelist files are included in the NU-WRF package in scripts/rip/.

6.3 ARWPOST

The ARWPOST program is a post-processor developed by NCAR for converting WRF-ARW netCDF data into GrADS format. Analogous to RIP, ARWPOST supports derivation of certain variables from the model output, and interpolation of those fields to isobaric or constant height levels. GrADS (see http://www.iges.org/grads) can then be used to visualize the data. The program is compiled by typing ./build.sh arwpost. Instructions for running ARWPOST are given in Chapter 9 of NCAR (2016).

6.4 UPP

The UPP program is the “Unified Post-Processor” developed by NOAA/NCEP for processing all NCEP model data. As with RIP and ARWPOST, UPP can read WRF netCDF output files, derive a number of meteorological fields from the provided model data, and interpolate to user specified levels. In the case of UPP, the data are output in GRIB format. The program is compiled by typing ./build.sh upp. Instructions for running UPP are given DTC (2015b).

The NU-WRF version of UPP includes several modifications provided by NASA SPoRT. These are experimental severe weather diagnostics:

- **Instantaneous Lighting Threat 1**: Based on grid-resolved graupel flux at -15C. Specified as “LIGHTNING THREAT 1” in parm/wrf.cntrl.parm file.
6.5 MET

- **Instantaneous Lightning Threat 2.** Based on vertically integrated ice. Specified as “LIGHTNING THREAT 2” in `parm/wrf_cntrl.parm` file.

- **Instantaneous Lightning Threat 3.** Based on Threat 1 and 2 products. Specified as “LIGHTNING THREAT 3” in `parm/wrf_cntrl.parm` file.

- **Interval Maximum Lighting Threat 1.** Based on grid-resolved graupel flux at -15°C. Specified as “MAX LGT THREAT 1” in `parm/wrf_cntrl.parm` file.

- **Interval Maximum Lighting Threat 2.** Based on vertically integrated ice. Specified as “MAX LGT THREAT 2” in `parm/wrf_cntrl.parm` file.

- **Interval Maximum Lighting Threat 3.** Based on Threat 1 and 2 products. Specified as “MAX LGT THREAT 3” in `parm/wrf_cntrl.parm` file.

In addition, UPP has been modified to read both graupel and hail mixing ratios, which are provided by the 4ICE microphysics scheme. Since the NCEP GRIB2 table does not list “hail” as a variable, hail is added to graupel for output and for use in visibility or radiance calculations. Also, UPP will read in radar reflectivity from the WRF output (variable `REFL_10CM`) if the 3ICE or 4ICE microphysics scheme was used, and this field will be used for any requested reflectivity product (in contrast, UPP calculates reflectivity for other microphysics schemes).

6.5 MET

The MET software is a community meteorological verification toolkit developed by the DTC. This is a generic tool for comparing gridded model forecasts and analyses against numerous observations – METARs, Mesonets, rawinsondes, MODIS satellite data, and Air Force cloud analysis data. MET expects the model data to be in GRIB format, a requirement that forces the user to run UPP on the WRF output first (see Section 6.4). Observation data formats include PREPBUFR and MADIS. With this input data, MET can be used for a number of different meteorological verifications, including point-to-point verification, object-oriented verification, and wavelet verification. Numerous statistical measures can be calculated with confidence intervals, and plotting capabilities are available.

The MET software is compiled by typing `./build.sh met`. Thorough instructions on running the software are provided in DTC (2015a).

6.6 LVT

LVT is a NASA developed land surface verification toolkit. It is designed to compare LIS output files against numerous in-situ, remotely sensed, and re-analysis products. Fields that can be evaluated include surface fluxes, soil
6.6 LVT

moisture, snow, and radiation. Multiple verification metrics can be calculated,
and advanced features include data masking, time series, temporal averaging,
and analysis of data assimilation impacts. The software is compiled by typing
`./build.sh lvt`. Detailed instructions on running LVT can be found in NASA
(2015c).
A Frequently Asked Questions

Q: What modules should I use on Discover and Pleiades when running NU-WRF?

A: Always load the identical modules that you used to compile NU-WRF. The current defaults on Discover are (assuming Bash shell):

```
source /usr/share/modules/init/sh
module purge
unset LD_LIBRARY_PATH
module load comp/intel-15.0.3.187
module load lib/mkl-15.0.3.187
module load other/comp/gcc-5.3-sp3 # Needed for Intel C++ compiler
module load mpi/sgi-mpt-2.12
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64
```

And on Pleiades:

```
source /usr/share/modules/init/sh
module purge
unset LD_LIBRARY_PATH
module load comp-intel/2015.3.187
module load gcc5/5.3.0 # Needed for Intel C++ compiler
module load mpi-sgi/mpt # Use generic SGI MPT module per NAS recommendation
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/lib64
```

If you use a non-default build configuration file, compare that file to the above and change the above modules accordingly. Likewise adjust the syntax if you are using a shell other than Bash.

The module settings should be done in your shell if you are running a program at the command line. If instead you are launching a batch job to SLURM or PBS, the above settings should be in the batch script, so the commands are run on the allocated compute node.

Q: What other environment settings should I use?

A: Make sure to set stack size to unlimited. This can help prevent memory allocation errors for automatic arrays. If using bash:

```
ulimit -s unlimited
```

If using csh:

```
limit stacksize unlimited
```
**Q:** What namelist settings should I use with WRF?

**A:** It is not possible to provide a single configuration optimal for all types of simulations (LES, regional climate, cloud system resolving, chemical transport, etc), but we have recommendations that provide a reasonable first-guess.

<table>
<thead>
<tr>
<th>Category</th>
<th>Selection</th>
<th>Namelist</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microphysics</td>
<td>NU-WRF Goddard 41CE</td>
<td>mp.physics=56</td>
<td>Latest stable NASA option.</td>
</tr>
<tr>
<td>Aerosol Coupling</td>
<td>GOCART simple aerosol</td>
<td>chem_opt=300, gsfgce_gocart_coupling=1, gsfrad_gocart_coupling=1, vertmix_onoff=1, chem_conv_tr=1, dust_opt=1, seas_opt=1, dmsemis_opt=1</td>
<td>Simple aerosol, coupled with radiation and microphysics, no gas chemistry.</td>
</tr>
<tr>
<td>LSM</td>
<td>LIS Noah 3.6</td>
<td>sf_surface.physics=55, num_soil_layers=4</td>
<td>Spin-up LIS on WRF grid for detailed initial fields; cannot use moving nests; must have lis.config file</td>
</tr>
<tr>
<td>PBL</td>
<td>MYNN2</td>
<td>bi_pbl.physics=5, sf_sfclay.physics=5</td>
<td>Replaces MYJ PBL scheme; used in RAPv2 and HRRR; reportedly gives unbiased PBL depth, moisture, and temperature.</td>
</tr>
<tr>
<td>Cumulus</td>
<td>G3</td>
<td>cu.physics=5, ishallow=1</td>
<td>Third-gen Grell scheme; tackles “grey zone”; compatible with used in RAPv2; handles shallow cumulus.</td>
</tr>
<tr>
<td>Category</td>
<td>Selection</td>
<td>Namelist</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Diffusion</td>
<td>2nd Order on coordinate surfaces; eddy</td>
<td>diff_opt=1</td>
<td>Appropriate for real data cases (dx $\geq$ 1 km)</td>
</tr>
<tr>
<td></td>
<td>coefficient based on deformation</td>
<td>km_opt=4</td>
<td></td>
</tr>
<tr>
<td>6th-order</td>
<td></td>
<td>diff_6th_opt=2</td>
<td>Removed 2*dx noise in light winds; can be tuned.</td>
</tr>
<tr>
<td>horizontal</td>
<td></td>
<td>diff_6th_factor=0.12</td>
<td></td>
</tr>
<tr>
<td>diffusion</td>
<td>Monotonic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advection</td>
<td>5th-order, positive-definite</td>
<td>moist_adv_opt=1</td>
<td>Used in RAPv2 and HRRR; positive-definite prevents negative mixing ratios from original non-negative values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>scalar_adv_opt=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tke_adv_opt=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>chem_adv_opt=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>momentum_adv_opt=1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>h_mom_adv_order=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>h_sca_adv_order=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>v_mom_adv_order=5</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>v_sca_adv_order=5</td>
<td></td>
</tr>
<tr>
<td>Rayleigh</td>
<td>Implicit</td>
<td>damp_opt=3</td>
<td>Prevents gravity waves from reflecting off model top; designed for real-data cases; used in RAPv2 and HRRR.</td>
</tr>
<tr>
<td>Damping</td>
<td></td>
<td>zdamp=5000.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>dampcoef=0.2</td>
<td></td>
</tr>
<tr>
<td>Vertical</td>
<td>Activated</td>
<td>w_damping=1</td>
<td>Damps updrafts approaching CFL limit; best used for long or quasi-operational runs.</td>
</tr>
<tr>
<td>Velocity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Damping</td>
<td>Tuning factor</td>
<td>epssm=0.1</td>
<td>Controls vertically-propagating sound waves; set to max slope of model terrain.</td>
</tr>
<tr>
<td>Time Off-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Centering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nesting</td>
<td>1-Way</td>
<td>feedback=0</td>
<td>2-way nesting does not work with LIS, and can lead to strange results with high-res nesting.</td>
</tr>
</tbody>
</table>

The above recommendations have some caveats:

- The PBL and cumulus settings are the most debatable physics selections.
  Our recommendations are mostly because of their use in the NCEP RAPv2
and HRRR models which speak to their robustness. In addition, WRF-Chem requires a Grell-family cumulus scheme for most simulations, placing these schemes at an advantage over popular alternatives such as Kain-Fritsch (Kain, 2004) and Betts-Miller-Janjić (Janjić, 1994). However, the G3 scheme has a pronounced wet bias when shallow cumulus is turned on, and the new Grell-Freitas scheme (Grell and Freitas, 2014) changes answers with varying CPU counts (the cause is under investigation). As for the PBL setting, the YSU scheme (Hong et al., 2006) is a popular alternative and includes options for subgrid orographic effects [e.g., Jiménez and Dudhia (2012)]. Users are encouraged to experiment.

• 6th-order diffusion was added to WRF because the normal diffusion scheme is tied to the wind speed, and can insufficiently smooth the fields in light winds (Knievel et al., 2007). Users running a short case may wish to turn off the 6th-order scheme to see if 2\( \Delta x \) features develop in the vertical velocity and divergence fields without it.

• A popular alternative to the positive-definite advection filter is the monotonic filter (Wang et al., 2009), which damps both positive and negative spikes in the advected fields (the positive-definite only damps negative spikes). Unfortunately monotonic advection may lead to excessive smoothing when 6th-order diffusion is also turned on. The user may wish to experiment with monotonic advection and turning off 6th-order diffusion, particularly with short chemistry runs where winds are not too light.

• Vertical velocity damping is artificial and is recommended mostly for situations where CFL violations are particularly unwelcome (e.g., quasi-operational runs).

• 2-way nesting cannot be used with WRF-LIS coupling because the feedback routine may change the land/sea mask for a parent WRF grid to better match the child WRF grid.

Q: What settings should I use with GEOGRID?

A: For GEOGRID, we recommend these settings:

• Use MODIS land use data instead of USGS. This requires changing the \texttt{geog\_data\_res} variable in \texttt{namelist.wps} to something like:
  \[
  \texttt{geog\_data\_res = 'modis\_30s+10m', 'modis\_30s+2m', 'modis\_30s+33s'},
  \]
  or, if using lake temperature initialization:
  \[
  \texttt{geog\_data\_res = 'modis\_lakes+10m', 'modis\_lakes+2m', 'modis\_lakes+33s'},
  \]
GEOGRID will check the `GEOGRID.TBL` settings to relate the selections to each dataset (terrain, soil type, etc). The ‘modis_30s’ (‘modis_lakes’) will only match with the land-use data and will force processing of MODIS data; the remaining data types will fall back on ‘10m’, ‘2m’, or ‘30s’ for the respective WRF grid. See Chapter 3 of [NCAR (2016)](http://www.nrel.gov/docs/fy16osti/64381.pdf) for more information.

- Process all EROD data with GEOGRID. This requires use of the new `GEOGRID.TBL.ARW_CHEM_NUWRF` table which lists entries for four different EROD datasets. The user does not need to decide which EROD option to use until running REAL.

**Q:** Why does WRF or REAL fail with this error about NUM\_LAND\_CAT?

---

**ERROR**

namelist : NUM\_LAND\_CAT = 20

input files : NUM\_LAND\_CAT = 24 (from geogrid selections).

---

**FATAL CALLED**

FATAL CALLED FROM FILE: (stdin) LINE: 709
Mismatch between namelist and wrf input files for dimension NUM\_LAND\_CAT

**A:** There are two possible causes.

First, you are running WRF without LIS coupling, your land use data is coming from GEOGRID, and your namelist settings for land use are inconsistent between GEOGRID, REAL, and WRF. Normally these programs expect USGS data (with 24 categories). If you configure GEOGRID to process MODIS instead, you must set NUM\_LAND\_CAT in namelist.input to 20 for consistency.

Second, you are trying to run WRF coupled with LIS; REAL is replacing the land use data from GEOGRID with that from LDT, and your namelist.input is not consistent. In this case, NUM\_LAND\_CAT should match the value from GEOGRID when you run REAL, but should match the value from LDT when you run WRF. (The reference to geogrid in the error message from WRF is incorrect in this case, and stems from the community WRF not knowing anything about LDT.) Note that LDT can provide USGS (24 categories), MODIS (20 categories), or UMD (14 categories).

**Q:** How do I convert between netCDF3 and netCDF4?

**A:** NetCDF comes with a NCCOPY utility that allows you to convert files between variants of netCDF. This can be useful for sharing NU-WRF netCDF4 data with outside users that prefer netCDF3, or for compressing large legacy netCDF3 files with netCDF4/HDF5. For best results, make sure you use a version of NCCOPY that was compiled with netCDF4/HDF5 compression support.

To convert to 64-bit netCDF3 (large file support):
nccopy -k 2 infile outfile

And to convert to netCDF4:

nccopy -k 3 infile output

It is also possible to remove compression from a netCDF4 file, which may make it readable to netCDF3 users:

nccopy -d 0 infile outfile

Run man nccopy for more information, including options for tuning netCDF4/HDF5 compression. Note that writing to netCDF3 classic format (-k 1) is not recommended due to file size limitations.

Q: To be asked.

A: To be answered.
B Porting NU-WRF

Currently NU-WRF is only fully supported for Intel compilers, for SGI MPT (on Discover and Pleiades), and for Intel MPI (on Discover). (We are also experimenting with GNU compilers and OpenMPI on Discover.) The underlying software should, however, run on other systems as long as the appropriate tools (compilers, MPI implementation, make, Perl, csh, bash, etc.) are available. Users who wish to port NU-WRF will need to take the following steps:

- Libraries:
  - Compile the libraries listed in section 4.2.
  - Determine the paths to the yacc binary and the flex library. Make sure yacc and flex are in your PATH environment variable.
  - Copy discover.cfg to a new top level build configuration file (here called newconfig.cfg, but any unique name can be used).
  - Edit newconfig.cfg to update the library paths.
  - Edit newconfig.cfg to change modules for system binaries and libraries (compilers, MPI, etc). If the Environmental Modules package [http://modules.sourceforge.net/] is not installed on your system, comment out the module commands and explicitly edit the PATH and LD_LIBRARY_PATH environment variables.

- ARWpost:
  - Inspect and edit ARWpost/arch/configure.defaults to ensure a block exists for the desired operating system, hardware, and compilers. Note that ARWpost is serial only (no MPI support).
  - Run ARWpost/configure at the command line to identify the integer value of the appropriate build selection.
  - Edit newconfig.cfg to enter the configure option as environment variable ARWPOST_CONFIGURE_OPT.
  - In the top NU-WRF directory, run ./build.sh --config newconfig.cfg arwpost to test the build.

- CASA2WRF:
  - Create a new makefile template in directory utils/casa2wrf/ to specify compilers and compiler flags.
  - Edit newconfig.cfg to enter the new makefile template name as environmental variable CASA2WRF_MAKEFILE. If necessary, also edit the CASA2WRF_PPROC_FC and CASA2WRF_PPROC_FFLAGS variables to specify compiler and compiler flags for the preprocessors.
In the top NU-WRF directory run `./build.sh --config newconfig.cfg casa2wrf` to test the build.

- **GEOS2WRF and MERRA2WRF:**
  - Create a new makefile template in directory `utils/geos2wrf/` to specify compilers and compiler flags.
  - Edit `newconfig.cfg` to enter the new makefile template name as environmental variable GEOS2WRF_MAKEFILE.
  - In the top NU-WRF directory, run `./build.sh --config newconfig.cfg geos2wrf` to test the build.

- **GOCART2WRF:**
  - Create a new makefile template in directory `utils/gocart2wrf/` to specify compilers and compiler flags.
  - Edit `newconfig.cfg` to enter the new makefile template name as environmental variables GOCART2WRF_MAKEFILE.
  - In the top NU-WRF directory, run `./build.sh --config newconfig.cfg gocart2wrf` to test the build.

- **GSDSU:**
  - Create a new makefile template in directory `GSDSU/SRC/` specifying the appropriate compilers, compiler flags, and MPI library if applicable.
  - Edit `newconfig.cfg` to enter the new makefile template name as environmental variable SDSU_MAKEFILE.
  - In the top NU-WRF directory, run `./build.sh --config newconfig.cfg gsdsu` to test the build.

- **LDT:**
  - Edit `ldt/arch/Config.pl` to specify the compiler flags. Compilers are specified using the LDT_ARCH environment flag (e.g., 'linux_ifc' indicates Intel compilers on Linux).
  - Edit `newconfig.cfg` to specify the correct LDT_ARCH value. If necessary, also edit the LDT_FC and LDT_CC variables to specify the correct MPI compiler wrappers.
  - In the top NU-WRF directory, run `./build.sh --config newconfig.cfg ldt` to test the build.

- **LISCONFIG:**
  - Create a makefile template in directory `utils/lisconfig/` to specify compilers and compiler flags.
– Edit `newconfig.cfg` to enter the new makefile template name as environmental variable `LISCONFIG_MAKEFILE`.

– In the top NU-WRF directory, run `./build.sh --config newconfig.cfg lisconfig` to test the build.

• LVT:

– Edit `lvt/arch/Config.pl` to specify the compiler flags. Compilers are specified using the LVT_ARCH environment flag (e.g., 'linux_ifc' indicates Intel compilers on Linux).

– Edit `newconfig.cfg` to specify the correct LVT_ARCH value. If necessary, also edit the LVT_FC and LVT_CC variables to specify the correct MPI compiler wrappers.

– In the top NU-WRF directory, run `./build.sh --config newconfig.cfg ldt` to test the build.

• MET:

– Edit `newconfig.cfg` to set the CXX, CC, and F77 environment variables, which defines the names of the C++, C, and Fortran compilers, respectively.

– In the top NU-WRF directory, run `./build.sh --config newconfig.cfg met` to test the build.

• NDVIBARENESS4WRF:

– Create a new makefile template in directory `utils/ndviBareness4Wrf/` to specify compilers and compiler flags.

– Edit `newconfig.cfg` to enter the new makefile template name as environmental variables `NB4W_MAKEFILE`.

– In the top NU-WRF directory, run `./build.sh --config newconfig.cfg ndviBareness4Wrf` to test the build.

• PREP_CHEM_SOURCES:

– Create a new makefile template in directory `utils/prep_chem_sources/bin/build/` to specify compilers and compiler flags. Note that the file name must use the naming convention `include.mk.*`.

– Edit `newconfig.cfg` to enter the suffix of the new Makefile template name as environmental variable `MAKEPSC_OPT`.

– In the top NU-WRF directory, run `./build.sh --config newconfig.cfg prep_chem_sources` to test the build.

• RIP4:

– Inspect and edit `RIP4/arch/configure.defaults` to ensure a block exists for the desired operating system, hardware, and compilers. Note that RIP4 is serial only (no MPI support).
- Run **RIP4/configure** at the command line to identify the integer value of the appropriate build selection.
- Edit **newconfig.cfg** to enter the configure option as environmental variable **RIP_CONFIGURE_OPT**.
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg rip** to test the build.

**SST2WRF**:

- Create a new makefile template in directory **utils/sst2wrf/** to specify compilers and compiler flags.
- Edit **newconfig.cfg** to enter the makefile template name as environmental variable **SST2WRF_MAKEFILE**.
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg sst2wrf** to test the build.

**UPP**:

- **NOTE**: Make sure WRFV3 is ported first.
- Inspect and edit **UPP/arch/configure.defaults** to ensure a block exists for the desired operating system, hardware, compilers, and MPI implementation.
- Run **UPP/configure** at the command line to identify the integer value of the appropriate build selections.
- Edit **newconfig.cfg** to enter the configure option as environmental variable **UPP_CONFIGURE_MPI_OPT**.
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg upp** to test the build.

**WPS**:

- **NOTE**: Make sure WRFV3 is ported first.
- Inspect and edit **WPS/arch/configure.defaults** to ensure a block exists for the desired operating system, hardware, compilers, and MPI implementation.
- Run **WPS/configure** at the command line to identify the integer value of the appropriate build selection.
- Edit **newconfig.cfg** to enter the configure option as environmental variable **WPS_CONFIGURE_MPI_OPT**. If necessary, also edit the **WPS_DEBUG_CFLAGS**, **WPS_DEBUG_FFLAGS**, and **WPS_DEBUG_F77FLAGS** variables to specify appropriate debugging flags for your compilers.
- In the top NU-WRF directory, run **./build.sh --config newconfig.cfg wps** to test the build.

**WRFV3 and LIS**
- Inspect and edit `WRFV3/arch/configure_new.defaults` to ensure a block exists for the desired operating system, hardware, compilers, and MPI implementation.

- Run `WRFV3/configure` to identify the integer value of the appropriate build selection.

- Create a new `configure.lis` makefile template in `WRFV3/lis/arch/` with appropriate compiler selection. NOTE: This approach is used instead of running the LIS `configure` script because LDFLAGS must be absent from the `configure.lis` file if the LIS code is compiled for coupling; also, it is easier to pass consistent debugging compiler flags to WRF and LIS by having the NU-WRF build system do it on the fly.

- Edit the `newconfig.cfg` to enter the configure option as environmental variable `WRF_CONFIGURE_MPI_OPT`. Also list the makefile template with environmental variable `WRF_CONFIGURE_LIS_MPI`. Also, set environmental variable `LIS_ARCH` to the value appropriate for your operating system and compiler [see NASA (2015b) for options]. If necessary, also edit the `WRF_DEBUG_CFLAGS_LOCAL` and `WRF_DEBUG_FCOPTIM` variables to specify appropriate debugging flags for your compilers.

- In the top NU-WRF directory, run `./build.sh --config newconfig.cfg wrf` to test the build. Also test with the chem and kpp targets.
References


REFERENCES


REFERENCES


