The WRF Standard Initialization (WRF SI)

John Smart

NOAA / Forecast Systems Laboratory

June 28, 2004

WRF Standard Initialization
What does it do?

- *Defines 3D WRF domains and nests.*
- *Generates the associated lats, lons and interpolates terrain, landuse, and other static land-state data onto these domains.*
- *Processes 3D background model data onto the 3D WRF gridded domains*

WRF Standard Initialization

Role of FSL

- Agency leading the SI Working Group
- Maintain SI Web page keeping current working versions; presently WRFSI_v2.0.1.
- Provide maintenance and enhancements to SI system. Direction also from WRF WG-14 and user community.
- Real time LAPS Hot Start initialization capability (work by Brent Shaw)

WRFSI Current Capabilities

- ***Provides all required initial and time-varying boundary conditions required to produce a forecast on the native WRF coordinate***
- ***Provides a method to define and localize a WRF domain, nests, and subnests***
  - *Location, projection type, dimensions, resolution*
  - *Acquisition/interpolation of "static" fields; nest attributes in static files.*
  - *Ingest readily-available gridded fields (NCEP GriB)*
  - *Horizontally/vertically interpolate to WRF domain*
  - *Grid staggering to support dynamic core*
  - *Derivation of state variables required by dynamic core*
  - *Consistency of LSM variables for WRF LSM input*
- ***Capabilities are available in the WRF Graphical User interface (Demo on Tue @ 8:30 by Paula McCaslin)***

WRFSI Software Design

- Modular
  - Components can be run individually
  - New components can be easily inserted into run stream
  - New options can be more easily added
  - Single namelist file for control (static/wrfsi.nl)
- Maintainable
  - CVS Revision control
  - Multiple platform build and testing strategy
  - Design emphasis on portability

7 easy steps for WRFSI installation and setup

1. Get WRFSI_v2.0.1 and the geographic data
2. Untar WRFSI_v2.0.1 (eg., /scratch/lapb/smart/wrf/wrfsi; aka $SOURCE_ROOT)
3. Run install_wrfsi.pl (eg., /scratch/lapb/smart/wrf/wrfsi; aka $INSTALLROOT)
   - Note: it is optional to install the GUI; that is, the install script will ask you.
4. Define domains and create nests if necessary (eg.,/scratch/lapb/smart/wrf/domains/tutorial; aka $DATAROOT)
5. Localize; use window_domain_rt.pl or GUI to obtain backwards compatability.
6. Prepare (degrib and reformat) background model information (grib_prep.pl)
7. Generate domain specific initial and lateral bndy condition files (wrfprep.pl)

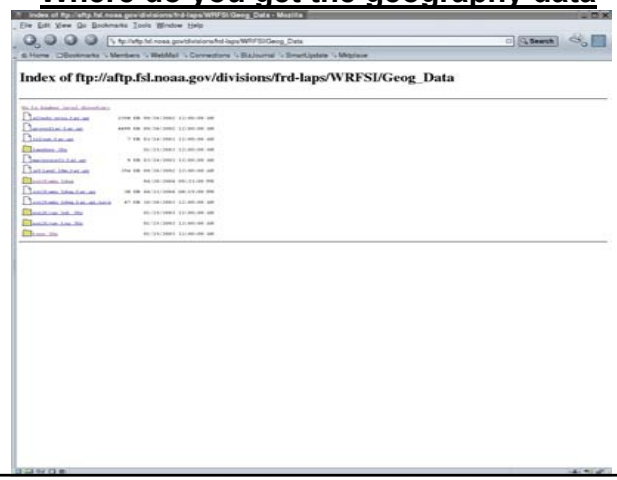
Required WRFSI Files

http://www.wrf-model.org/si/release/wrfsi_v2.0.1.tar.gz (~30mb)

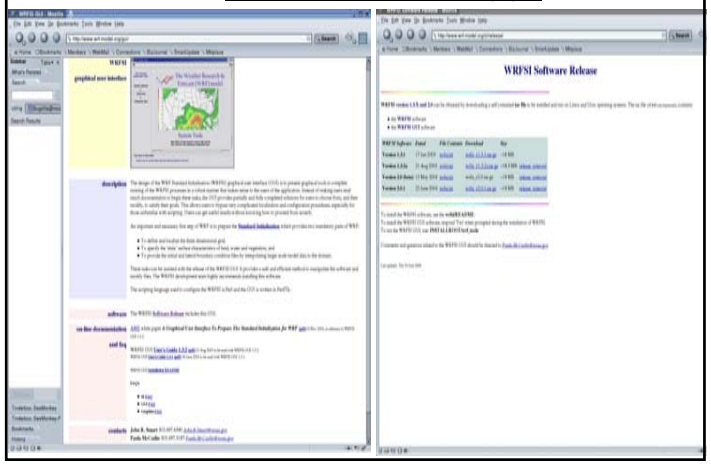
Static Geographical Data (large ones in quartersphere files).
Available for download at FSL ftp site:
ftp://aftp.fsl.noaa.gov/divisions/frd-laps/WRFSI/Geog_data

- Topo_30s/ (data in 10x10 deg tiles; eg., 30N120W)
  - 1) topo_30s_SW.tar.gz, 2) _SE.tar.gz, 3) _NW.tar.gz, 4) _NE.tar.gz.
- Landuse_30s/
  - landuse_30s_NW.tar.gz, 2) _SE.tar.gz, 3) _NW.tar.gz, 4) _NE.tar.gz.
- Green fraction: greenfrac.tar.gz (see directory /soiltemp_1deg)
- ***NEW***: Soiltemp: soiltemp_1deg/soiltemp_1deg.tar.gz (two 180 tiles)
- Soiltype (top layer): soiltype_bot_30s (NW, NE, SW, SE)
- Soiltype (bottom layer): soiltype_bot_30s (NW, NE, SW, SE)
- Albedo: albedo_ncep: albedo.tar.gz
- Max Snow Albedo: maxsnowalb.tar.gz
- Terrain Slope Index: islope.tar.gz

Where do you get the geography data



Where do you get WRFSI_v2.0.1?
http:/www.wrf-model.org/gui



Required External Packages

- Unix or Linux OS
- Fortran 90 Compiler
- C Compiler (gcc is sufficient)
- make Utility
- Perl
- netCDF Libraries

WRFSI Setup

- Compiling
  - Uses Unix or Gnu make utility
    - Machine-dependent settings in one shared file ($SRCROOT/src/include/makefile.inc)
    - Top-level makefile ($SRCROOT/Makefile) can compile and install everthing, or each component can be individually compiled.
  - Perl script "install_wrfsi.pl" handles editing of makefile.inc file, compilation, and installation based on command-line options
    - Requires $SRCROOT/src/include/makefile_{machine}.inc.in to be present with correct compiler options, etc.

After downloading wrfsi_v2.0.1

/scratch/lapb/smart/wrf > ls -rtal

-rw------- 1 smart frdlapb 30597120 Jun 24 16:34 wrfsi_v2.0.1.tar
drwxrwsr-x 8 smart frdlapb 20480 Jun 24 16:44 ../
drwxrwsr-x 2 smart frdlapb 4096 Jun 24 16:45 ./

After untarring wrfsi v2.0.1

```
homer:/scratch/lapb/smart/wrf > tar -xf wrfsi_v2.0.1.tar
homer:/scratch/lapb/smart/wrf > ls -rtal

drwxr-sr-x    9 smart    frdlapb      4096 Jun 22 13:49 wrfsi/
-rw-------    1 smart    frdlapb  30597120 Jun 24 16:34 wrfsi_v2.0.1.tar
drwxrwsr-x    8 smart    frdlapb     20480 Jun 24 16:44 ../
drwxrwsr-x    3 smart    frdlapb      4096 Jun 24 16:50 ./

homer:/scratch/lapb/smart/wrf> cd wrfsi

homer:/scratch/lapb/smart/wrf/wrfsi > ls -rtal

-rw-r--r--    1 smart    frdlapb      4405 Jan 17  2003 INSTALL
-rw-r--r--    1 smart    frdlapb     11166 Mar 11  2003 HOW_TO_RUN.txt
-rw-r--r--    1 smart    frdlapb     30463 Apr 22  2003 README
-rw-r--r--    1 smart    frdlapb      4101 Dec 10  2003 Makefile
-rwxr-xr-x    1 smart    frdlapb     25303 May 25 12:02 install_wrfsi.pl*
-rw-r--r--    1 smart    frdlapb     12664 Jun 18 13:35 README.wrfsi_nl
-rw-r--r--    1 smart    frdlapb     15975 Jun 18 13:35 CHANGES
drwxr-sr-x    7 smart    frdlapb      4096 Jun 22 13:49 data/
drwxr-sr-x    2 smart    frdlapb      4096 Jun 22 13:49 etc/
drwxr-sr-x    3 smart    frdlapb      4096 Jun 22 13:49 graphics/
drwxr-sr-x    7 smart    frdlapb      4096 Jun 22 13:49 extdata/
drwxr-sr-x    5 smart    frdlapb      4096 Jun 22 13:49 gui/
drwxr-sr-x    3 smart    frdlapb      4096 Jun 22 13:49 util/
drwxr-sr-x   12 smart    frdlapb      4096 Jun 22 13:49 src/
```

- setenv SOURCE_ROOT /scratch/lapb/smart/wrf/wrfsi
- setenv INSTALLROOT /scratch/lapb/smart/wrf/wrfsi
- setenv DATAROOT /scratch/lapb/smart/wrf/domains
- setenv GEOG_DATAROOT /data/lapb/geog
- setenv EXT_DATAROOT /scratch/lapb/smart/wrf/extdata
- setenv TEMPLATES /scratch/lapb/smart/wrf/templates
- setenv NETCDF /usr/local/apps/netcdf
- setenv PATH_TO_PERL /usr/bin/perl

INSTALLING WRFSI

```
homer:/scratch/lapb/smart/wrf/wrfsi > perl install_wrfsi.pl
No path to MPICH system found!
--path_to_netcdf not specified, attempting to determine...
netCDF path found from environment variable.

Do you want to install the WRF SI graphical user interface? [y|n]:
--machine_type not specified, attempting to determine...
Compiling for pc
Directory setup:
SRCROOT = /scratch/lapb/smart/wrf/wrfsi
INSTALLROOT = /scratch/lapb/smart/wrf/wrfsi
DATAROOT = /scratch/lapb/smart/wrf/domains
EXT_DATAROOT = /scratch/lapb/smart/wrf/extdata
GEOG_DATAROOT = /data/lapb/geog
TEMPLATES = /scratch/lapb/smart/wrf/templates
NETCDF = /usr/local/apps/netcdf
Make and install wrfsi binaries in /scratch/lapb/smart/wrf/wrfsi
```

INSTALLING WRFSI (Cont)

```
Check installation in /scratch/lapb/smart/wrf/wrfsi/etc
-rwxrwxr-x    1 smart    frdlapb      6933 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/wrfsi.pl
-rwxrwxr-x    1 smart    frdlapb     34495 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/wrfprep.pl
-rwxrwxr-x    1 smart    frdlapb     26635 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/window_domain_rt.pl
-rwxrwxr-x    1 smart    frdlapb      8755 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/sync_wrfnl.pl
-rwxrwxr-x    1 smart    frdlapb       613 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/setup_mpiconf.pl
-rwxrwxr-x    1 smart    frdlapb     10964 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/run_wrf.pl
-rwxrwxr-x    1 smart    frdlapb      9260 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/rt_wrf.pl
-rwxrwxr-x    1 smart    frdlapb     23478 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/localize_domain.pl
-rwxrwxr-x    1 smart    frdlapb      7511 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/laps_driver.pl
-rwxrwxr-x    1 smart    frdlapb     11151 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/grib_prep218.pl
-rwxrwxr-x    1 smart    frdlapb     19342 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/grib_prep.pl
-rwxrwxr-x    1 smart    frdlapb     15798 Jun 25 08:01 /scratch/lapb/smart/wrf/wrfsi/etc/generate_images.pl
Check installation in /scratch/lapb/smart/wrf/wrfsi/bin
drwxr-sr-x   10 smart    frdlapb      4096 Jun 25 08:00 ..
-rwxr-xr-x    1 smart    frdlapb    767263 Jun 25 08:01 grib_prep.exe
-rwxr-xr-x    1 smart    frdlapb    772806 Jun 25 08:01 hinterp.exe
-rwxr-xr-x    1 smart    frdlapb   1338806 Jun 25 08:01 vinterp.exe
-rwxr-xr-x    1 smart    frdlapb    914573 Jun 25 08:01 siscan
-rwxr-xr-x    1 smart    frdlapb   1029199 Jun 25 08:01 staticpost.exe
-rwxr-xr-x    1 smart    frdlapb   1147483 Jun 25 08:01 gridgen_model.exe
```

INSTALLING WRFSI (Cont)

```
homer:/scratch/lapb/smart/wrf/wrfsi > ls -rtal
-rw-r--r--   1 smart   frdlapb    4405 Jan 17  2003 INSTALL
-rw-r--r--   1 smart   frdlapb   11166 Mar 11  2003 HOW_TO_RUN.txt
-rw-r--r--   1 smart   frdlapb   30463 Apr 22  2003 README
-rw-r--r--   1 smart   frdlapb    4101 Dec 10  2003 Makefile
-rwxr-xr-x   1 smart   frdlapb   25303 May 25 12:02 install_wrfsi.pl*
-rw-r--r--   1 smart   frdlapb   12664 Jun 18 13:35 README.wrfsi_nl
-rw-r--r--   1 smart   frdlapb   15975 Jun 18 13:35 CHANGES
drwxr-sr-x   7 smart   frdlapb    4096 Jun 22 13:49 data/
drwxr-sr-x   3 smart   frdlapb    4096 Jun 22 13:49 graphics/
drwxr-sr-x   7 smart   frdlapb    4096 Jun 22 13:49 extdata/
drwxr-sr-x   3 smart   frdlapb    4096 Jun 22 13:49 util/
drwxr-sr-x  12 smart   frdlapb    4096 Jun 22 13:49 src/
-rw-rw-r--   1 smart   frdlapb   61169 Jun 25 08:01 make_install.log
drwxr-sr-x   2 smart   frdlapb    4096 Jun 25 08:01 etc/
drwxrwsr-x   2 smart   frdlapb    4096 Jun 25 08:01 bin/
-rw-rw-r--   1 smart   frdlapb    1062 Jun 25 08:01 config_paths
drwxrwsr-x   6 smart   frdlapb    4096 Jun 25 08:01 ../
lrwxrwxrwx   1 smart   frdlapb      13 Jun 25 08:01 wrf_tools -> gui/wrf_tools*
drwxr-sr-x   6 smart   frdlapb    4096 Jun 25 08:01 gui/
drwxr-sr-x  10 smart   frdlapb    4096 Jun 25 08:01 ./
```

INSTALLING WRFSI (Cont)

homer:/scratch/lapb/smart/wrf/wrfsi > cd ..

/scratch/lapb/smart/wrf

homer:/scratch/lapb/smart/wrf > ls -rtal

```
-rw-------   1 smart   frdlapb  30597120 Jun 24 16:34 wrfsi_v2.0.1.tar
drwxr-sr-x   7 smart   frdlapb      4096 Jun 25 08:00 extdata/
drwxrwsr-x   2 smart   frdlapb      4096 Jun 25 08:00 domains/
drwxrwsr-x   3 smart   frdlapb      4096 Jun 25 08:01 templates/
drwxrwsr-x   6 smart   frdlapb      4096 Jun 25 08:01 ./
drwxr-sr-x  10 smart   frdlapb      4096 Jun 25 08:01 wrfsi/
```

Program gridgen_model

- Run once when domain and nests initially defined. Uses environment variable FORCE_LOCALIZATION (= domain #, or = all).
- Acquires/derives all required static fields
  - Topography with user-controllable filtering parameters
  - Latitudes, longitudes, map factors, coriolis, etc. on model stagger (Arakawa C). Also A-grid for 3DVAR.
  - All required fields for LSM support
    - USGS 30 sec (~1km) topography (topo_30s)
    - USGS 24-category landuse (landuse_30s)
    - WMO/FAO 16-category 2-layer soil types (soiltype_top_30s and soiltype_bot_30s)
    - Annual mean deep soil temperature (soiltemp_1deg)
    - Monthly greenness fraction (greenfrac; 0.14 deg)
    - Albedo (Albedo_ncep; mean monthly values; 0.14 deg)
    - Max Snow Albedo (maxsnowalb; 1 deg)
    - Terrain slope index (islope; 1 deg)
  - Reads “tiled” data sets (available on FTP server)
- Output in netCDF file (static.wrfsi.d##)

Localizing a domain using window_domain_rt.pl

homer:/scratch/lapb/smart/wr/wrfsi > cd etc

/scratch/lapb/smart/wrf/wrfsi/etc

homer:/scratch/lapb/smart/wr/wrfsi/etc > ls -rtal window_domain_rt.pl

-rwxrwxr-x   1 smart   frdlapb   26635 Jun 25 08:01 window_domain_rt.pl*

homer:/scratch/lapb/smart/wrf/wrfsi/etc > setenv MOAD_DATAROOT /scratch/lapb/smart/wrf/domains/tutorial

homer:/scratch/lapb/smart/wrf/wrfsi/etc > cd ../../templates/

/scratch/lapb/smart/wrf/templates

homer:/scratch/lapb/smart/wrf/templates > mkdir tutorial

homer:/scratch/lapb/smart/wrf/templates > cd tutorial

/scratch/lapb/smart/wrf/templates/tutorial

homer:/scratch/lapb/smart/wrf/templates/tutorial > cp ../../wrfsi/data/static/wrfsi.nl .

homer:/scratch/lapb/smart/wrf/templates/tutorial > vi wrfsi.nl

Localizing a domain using window_domain_rt.pl

```
&hgridspec
NUM_DOMAINS = 1
XDIM = 100
YDIM = 67
PARENT_ID = 1
RATIO_TO_PARENT = 1
DOMAIN_ORIGIN_LLI = 1
DOMAIN_ORIGIN_LLJ = 1
DOMAIN_ORIGIN_URI = 100
DOMAIN_ORIGIN_URJ = 67
MAP_PROJ_NAME = 'lambert'
MOAD_KNOWN_LAT = 40.68
MOAD_KNOWN_LON = -73.80
MOAD_STAND_LATS = 40.68, 40.68
MOAD_STAND_LONS = -73.80
MOAD_DELTA_X = 14540
MOAD_DELTA_Y = 14540
SILAVWT_PARM_WRF = 0.
TOPTWVL_PARM_WRF = 2.
/
```

Localizing a domain using window_domain_rt.pl

```
&sfcfiles
TOPO_30S = '/scratch/lapb/smart/wrf/geog/topo_30s',
LANDUSE_30S = '/scratch/lapb/smart/wrf/geog/landuse_30s',
SOILTYPE_TOP_30S = '/scratch/lapb/smart/wrf/geog/soiltype_top_30s',
SOILTYPE_BOT_30S = '/scratch/lapb/smart/wrf/geog/soiltype_bot_30s',
GREENFRAC = '/scratch/lapb/smart/wrf/geog/greenfrac',
SOILTEMP_1DEG = '/scratch/lapb/smart/wrf/geog/soiltemp_1deg',
ALBEDO_NCEP = '/scratch/lapb/smart/wrf/geog/albedo_ncep',
MAXSNOWALB = '/scratch/lapb/smart/wrf/geog/maxsnowalb',
ISLOPE = '/scratch/lapb/smart/wrf/geog/islope',
```

Localizing a domain using window_domain_rt.pl

```
cd ../../wrfsi/etc
/scratch/lapb/smart/wrf/wrfsi/etc
homer:/scratch/lapb/smart/wrf/wrfsi/etc >perl window_domain_rt.pl -t
    /scratch/lapb/smart/wrf/templates/tutorial -w wrfsi -c
```

```
Run /scratch/lapb/smart/wrf/wrfsi/bin/staticpost.exe
/scratch/lapb/smart/wrf/wrfsi/bin/staticpost.exe > /scratch/lapb/smart/wrf/domains/tutorial/log/staticpost.log
    completed

Routine: sync_wrfnl.pl
MOAD_DATAROOT = /scratch/lapb/smart/wrf/domains/tutorial
/usr/bin/perl /scratch/lapb/smart/wrf/wrfsi/etc/sync_wrfnl.pl /scratch/lapb/smart/wrf/domains/tutorial
    completed

Checking for static file
checking localize_domain.log
-rw-rw-r--    1 smart    frdlapb     67769 Jun 28 06:51
    /scratch/lapb/smart/wrf/domains/tutorial/log/localize_domain.log
/scratch/lapb/smart/wrf/wrfsi/etc/localize_domain.pl complete
update prod subdirectory - synchronize with repository

 log file: /scratch/lapb/smart/wrf/domains/tutorial/log/localize_domain.log.28-Jun-04-06:58-AM
      ************************************************
      *******  Domain Localization complete ******
      ************************************************
```

WRFSI Nesting Setup

- The following namelist (wrfsi.nl) variables define nests:
- &interp_control (hinterp/vinterp control);
  - num_active_subnests = 0 if no nests are to be processed by h/v interp, only MOAD. If > 0 then it is the number of nests that will be processed by h/v interp
  - active_subnests = the domain number of the active subnest to be processed
- &hgridspec (gridgen_model input):
  - num_domains = Number of domains including the MOAD (=.d01).
  - parent_id = domain # of the nest (= 1 if no nest; =1,1 if one nest, =1,1,2 if you have a nests within a nest and num_domains = 3, and so forth).
  - ratio_to_parent = The number of grid points added between parent grid points minus 1. Eg., 2::1 means 1 extra grid point between parent grid points; 3::1 means 2 extra grid points between parent grid points, and so forth.
  - domain_origin [Ili/Ilj/uri/urj] = lower left and upper right integer i/j pairs of parent corresponding to nest.
  - Nest attributes: center lat/lon, xdim/ydim, and grid spacing are calculated using ratio_to_parent and MOAD xdim/ydim. Nest attributes are written to static file.

Program grib_prep

- Reads and reformats GRIB-format data files; More discussion is coming in just a minute!

Program hinterp

- Runs once per domain per model run
- Reads static file and grib_prep output
- Horizontally interpolates pressure level data to the WRF horizontal staggered (Arakawa-C) domain
- Utilizes special routines for interpolation of masked fields (e.g., soil moisture, soil temperature, SST, snowcover, etc.)
- Selectable interpolation method (wrfsi.nl namelist variables “hinterp_method” and “lsm_hinterp_method”
- Processes any variable contained in grib_prep output
- Outputs binary-format data and metadata

Program vinterp

- Run once per domain per simulation
- Vertically interpolates pressure levels to WRF coordinate
- Derives variables needed for input to WRF dynamic engine
- Processes all variables found in hinterp output
  - “Pass-through” of scalars, 1D, and 2D variables
  - Vertical interpolation of 3D variables
- Performs final staggering of atmospheric variables
- Outputs AP I/O for input into the WRF "real.exe" initialization program

Perl Scripts

- install_wrfsi.pl: Builds the software
- window_domain_rt.pl: Sets up domains from pre-existing namelist templates
  - localize_domain.pl: Edits namelists and cdl file and runs gridgen_model.exe
- *sync_wrfnl.pl: synchronizes the wrf.nl and wrfsi.nl in $MOAD_DATAROOT insuring that they have the same dimensions, nests, and grid spacing.*
- *generate_images.pl: Uses NCL scripts to generate land-state (terrain, landuse, etc.) images (eg., $MOAD_DATAROOT/static/avc.ncgm).*

Miscellaneous Utilities

- IDL Routines
  - read_wrf_static.pro: Reads WRFSI static file
  - read_gribprep_output.pro: Read grib_prep output files
  - get_wrf_variable.pro: Reads hinterp/vinterp output
- Program staticpost.exe
  - Reformats static.wrfsi.d## (produces new file called wrfstatic_d##) for ingest by real.exe
- Program siscan
  - Dumps a summary of wrfsi file contents
- Now let's take a close look at the grib_prep.pl and wrfprep.pl functions for initializing a WRF domain.

WRFSI Setup

- Directory structure and environment variables
  - "Source root" ($SRCROOT) -- Approx. 10 MB
    - Typically this is the top-level directory created after extracting the tar file
    - Contains src, etc, gui, graphics,and data subdirectories
  - "Installation root" ($INSTALLROOT) -- Approx. 10 MB
    - Where installed code will reside
    - Contains bin and etc subdirectories
  - "MOAD data root" ($MOAD_DATAROOT)
    - Where data specific to a single domain resides
    - Contains static, siprd, wrfprd, cdl, and log subdirectories
  - "External data root" ($EXTDATAROOT)
    - Location of grib_prep output
    - Contains extprd, work, and log subdirectories

Plans

(as resources allow)

Support for 2-way nested domains.

Graphical User Interface enhancements.

- Accommodate other initializations (3DVAR, LAPS, etc.).
- Schedule/manage model runs.

Parallelization: some progress made in (03-04).

SI to help support WRF chemistry modeling.

Maintenance and general software improvements.

- Using the software is getting easier.

Initializing a WRF Domain

- Assumes "localization" is already done
- Includes:
  - Data formatting for input to hinterp
    - "grib_prep" for GRIB1 data
    - User programs for other formats
    - Intermediate format defined in README, similar to MM5 pregrid output
  - Horizontal and vertical interpolation
  - Grid staggering
  - Output to WRF netCDF format via IO/API for input to real.exe
  - Scripts to drive the programs, including optional running of real.exe

Grib_prep Preprocessor

- Extracts variables from GRIB-1 format data
  - Variable extraction defined in Vtable files found in EXT_DATAROOT/static
- Outputs intermediate binary format files for ingest by hinterp.exe
- Can perform temporal interpolation if needed

Grib_prep Preprocessor

- Vtables already provided for AVN/GFS, Eta, NNRP, RUC (isobaric) and RUC (hybrid coord)
- Program controlled by EXT_DATAROOT/static/grib_prep.nl
- Program executed by INTALLROOT/etc/grib_prep.pl

grib_prep Setup

- Ensure EXT_DATAROOT was set up properly during installation of SI
- Edit paths to GRIB data in EXT_DATAROOT/static/grib_prep.nl
- Edit names and cycle info of data sources used if necessary
- Run for each new cycle of Eta, GFS, or whatever input data you use, not necessarily for each WRF domain! For example, one run of the global GFS model can support multiple WRF domains around the world. Grib_prep output is domain-independent (no spatial interpolation done)

grib_prep.nl: Key Items

- SRCNAME: Array of character strings assigning the name (prefix) used for a particular data source (e.g., "GFS", "ETA")
- SRCVTAB: Defines suffix of Vtable to use for each SRCNAME
- SRCPATH: Path to each SRCNAME GRIB files
- SRCCYCLE: Number of hours between cycles for each SRCNAME (e.g., ETA is a 6-hourly cycle, so SRCCYLE would be 6)
- SRCDELAY: Number of hours after cycle time when a complete run for each SRCNAME is available to you.

grib_prep.nl

- Time info edited by grib_prep.pl
  - If no time options specified, then grib_prep.pl assumes latest available cycle for SOURCE is desired, based on current UTC time, SRCCYCLE, and SRCDELAY.
  - Specific cycle times, forecast length, output interval, etc. can be specified on command line

Running grib_prep.pl

- Syntax: grib_prep.pl [options] SOURCE
  - SOURCE must be one of the items listed in SRCNAME in grib_prep.nl
  - Any options specified must be provided on command line *before* SOURCE or they will be ignored

Grib_prep: Common Options

- Time control options:
  - "-l *n"*: Attempts to process data files to cover *n* hours of integration time
  - "-t *n"*: Sets the interval between data files to *n* hours. Grib_prep will temporally interpolate if necessary
  - "-s *YYYYMMDDHH"*: Sets cycle time of SOURCE to UTC date/time specified
  - "-e YYYYMMDDHH": Sets end time of data to be processed

Grib_prep: Common Options

- Miscellaneous
  - "-f *filter"*: Used to specify a Perl regex pattern match to use when searching source GRIB directory.
  - "-h": Displays all possible options
  - "-d": Sets or overrides the EXT_DATAROOT variable
  - "-P" : Causes older grib_prep output files in EXT_DATAROOT/extprd to be purged upon successful run (useful for real-time users)

Initializing a WRF Domain

- Driver: $INSTALLROOT/etc/wrfprep.pl (replaces wrfsi.pl as preferred run script)
- Programs called
  - $INSTALLROOT/bin/hinterp.exe
  - $INSTALLROOT/bin/vinterp.exe
  - $INSTALLROOT/../main/real.exe (optional)
- Operates on/in $MOAD_DATAROOT
  - Always processes Mother Of All Domains
  - Optionally creates initial output for nests
- Controlled by
  - $MOAD_DATAROOT/static/wrfsi.nl
  - $MOAD_DATAROOT/static/wrf.nl (if running real.exe)
  - wrfprep.pl command line options

Running wrfprep.pl

- Assumes you have already run grib_prep.pl or your own program to create data files ready for hinterp ingest, and that all required times are available (any temporal interpolation of input data must already be done!)
- Syntax: "wrfprep.pl [options]"
- If no options are set, wrfprep assumes:
  - MOAD_DATAROOT environment variable is set
  - Initialization time is current UTC hour
  - Forecast length is 24 hours
  - Time between LBC updates is correct in wrfsi.nl
- Use "wrfprep.pl –h" to see all valid options

What wrfprep.pl Does

- Edits namelists in MOAD_DATAROOT
  - Updates time entries in static/wrfsi.nl
  - If running real.exe (-r s|p option):
    - Copies static/wrf.nl to siprd/namelist.input
    - Edits times/dimensions of siprd/namelist.input
- Soft links time-appropriate input files into siprd
  - Source directories specified in si_paths section of wrfsi.nl
  - File name prefixes to look for specified in INIT_ROOT, LBC_ROOT, and LSM_ROOT entries of wrfsi.nl

Additional Info on Input Data

- For 1st time period (initial time), wrfsi uses the following priority order:
  - {CONSTANTS_PATH}/{CONSTANTS_FULL_NAME}
  - {LSMPATH}/{LSM_ROOT}:YYYY-MM-DD_HH
  - {ANALPATH}/{INIT_ROOT}:YYYY-MM-DD_HH
  - {ANALPATH}/{INIT_ROOT}:YYYY-MM-DD_HH
- For subsequent time periods:
  - {CONSTANTS_PATH}/{CONSTANTS_FULL_NAME}
  - {LSMPATH}/{LSM_ROOT}:YYYY-MM-DD_HH
  - {LBCPATH}/{INIT_ROOT}:YYYY-MM-DD_HH
- Minumum entries needed: INIT_ROOT and LBC_ROOT
  - They can be (and are often) the same
- LSM and CONSTANTS are optional files if you have LSM fields, SST data, etc. coming from a different source than your initial/lateral boundary conditions

What wrfprep Does (cont.)

- Executes hinterp and vinterp in MOAD_DATAROOT/siprd
- Executes real.exe in MOAD_DATAROOT/siprd (optional)
- Checks run success
- Creates a small text info file in siprd

Common wrfprep Options

- -d /your/moad_dataroot
  - Specifies MOAD_DATAROOT to set or override the MOAD_DATAROOT environment variable (useful when running wrfprep in cron)
- -s YYYYMMDDHH
  - Sets a specific UTC start date/time (useful for case studies
- -e YYYYMMDDHH
  - Sets a specific UTC end/date time (alternative to –f)

Common wrfprep Options

- -f *n*
  - Sets a run length equal to *n* hours (alt. to –e)
- -t *n*
  - Sets input data time interval to n hours
- -o *n*
  - Add *n* hours to current UTC time to set start time (useful for cron jobs). *n* must be negative if setting start time to a time previous to current time (e.g., to run a 12Z initialization at 14Z, *n* would be –2)
- -T
  - Prevents time editing...just uses entries already in wrfsi.nl

Running real.exe with wrfprep

- WRFSI INSTALLROOT must be a subdirectory of your WRFV2 directory
  - Assumes real.exe is in INSTALLROOT/../main
- If running MPI version of real.exe, you must set MPICH environment variable to path of MPICH installation when you run the install_wrfsi.pl script
  - e.g., setenv MPICH /usr/local/mpich
  - Assumes "mpirun" found in $MPICH/bin
- You must have a valid version of the WRF namelist.input file in MOAD_DATAROOT/static, and it must be named "wrf.nl"

Running real.exe with wrfprep

- Edit MOAD_DATAROOT/static/wrf.nl
  - Set physics options, delta-t, output options, etc.
  - wrfprep will handle start/stop time options
- Set the –r option on wrfprep.pl
  - Use "-r s" if you compiled real.exe as serial code
  - Use "-r p" if you compiled real.exe as MPI
    - Specify *n* processors to use by also adding "-p *n*"
    - Specify file listing available machines using MACHINE_FILE or GMPICONF environment variable, if your system does not do this already as part of a job queue system...or, you can place the machine file listing (appropriate for your version of MPICH) in MOAD_DATAROOT/static/mpi_machines.conf