WRF OVERVIEW AND COMPILING PROCESS

Kelly Werner / Dave Gill
NCAR/MMM
February 2019

INTRODUCTION TO WRF

- What is WRF?
  - Weather Research and Forecasting Model
  - Used for both research and operational forecasting
  - Supported community model
    - Free & shared resource with distributed development and centralized support
- Current Version: V4.0.3

WRF APPLICATIONS

- Research
  - Meteorological case-studies for various applications and scales
  - Regional climate and seasonal time-scale
  - Air quality, wind energy, hydrology, tropical cyclones, etc.
- Real-time numerical weather prediction (forecasting)
- Idealized simulations
- Data Assimilation (WRFDA)
- Chemistry applications (WRF-Chem)
- Hydrological coupling (WRF-Hydro)
- Wildfire Modeling (WRF-Fire)

Developed & supported by other groups

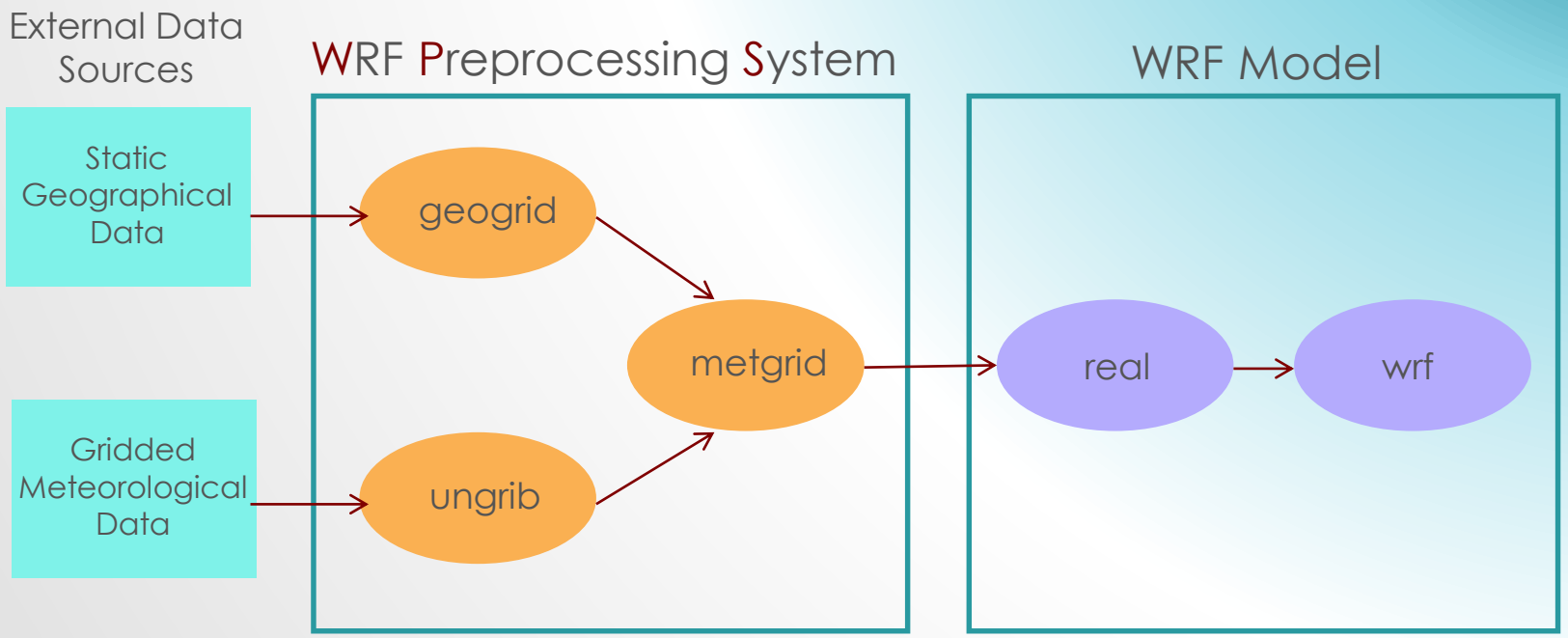
INTRODUCTION TO WPS

- What is WPS?
  - WRF Preprocessing System
  - Prepares input to WRF for real-data simulations
- Current Version: V4.0.3

WPS PROGRAMS & THEIR FUNCTIONS

- Geogrid
  - Defines simulation domain area
  - Produces terrain, landuse, soil type, etc. on the domain
- Ungrib
  - De-gribs GRIB-formatted meteorological input data files
  - Converts to intermediate format (expected by metgrid)
- Metgrid
  - Horizontally interpolates meteorological to WRF model grid

WPS/WRF FLOWCHART



COMPILING WRF/WPS

INSTALLING STEPS

- ***Check system requirements***
- Installing libraries
- Download source data
- Compile WRF
- Compile WPS
- Download initial/BC datasets

SYSTEM REQUIREMENTS

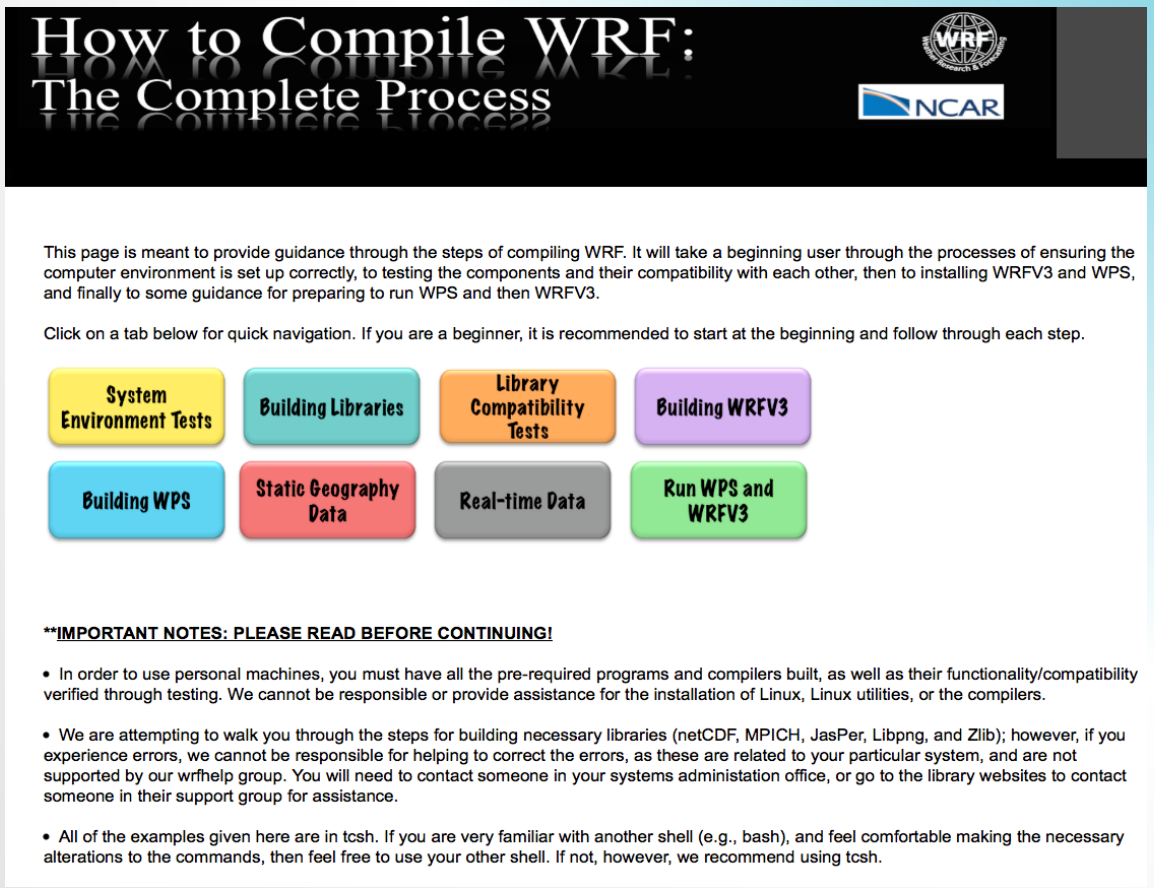
- On what kinds of systems will WRF run?
  - Generally any 32- or 64-bit hardware, running a UNIX-like operating system
  - You may also use dual-booting into a UNIX-like OS (e.g., Windows with Linux built parallel)

- Examples of acceptable systems:
  - Laptops, desktops, and clusters running Linux
  - Laptops and desktops running MacOS X
  - Clusters running Unix-like: Linux, AIX

CHECK SYSTEM REQUIREMENTS

- Webpage:

http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php



How to Compile WRF: The Complete Process

This page is meant to provide guidance through the steps of compiling WRF. It will take a beginning user through the processes of ensuring the computer environment is set up correctly, to testing the components and their compatibility with each other, then to installing WRFV3 and WPS, and finally to some guidance for preparing to run WPS and then WRFV3.

Click on a tab below for quick navigation. If you are a beginner, it is recommended to start at the beginning and follow through each step.

System Environment Tests | Building Libraries | Library Compatibility Tests | Building WRFV3

Building WPS | Static Geography Data | Real-time Data | Run WPS and WRFV3

**IMPORTANT NOTES: PLEASE READ BEFORE CONTINUING!

- In order to use personal machines, you must have all the pre-required programs and compilers built, as well as their functionality/compatibility verified through testing. We cannot be responsible or provide assistance for the installation of Linux, Linux utilities, or the compilers.
- We are attempting to walk you through the steps for building necessary libraries (netCDF, MPICH, JasPer, Libpng, and Zlib); however, if you experience errors, we cannot be responsible for helping to correct the errors, as these are related to your particular system, and are not supported by our wrfhelp group. You will need to contact someone in your systems administation office, or go to the library websites to contact someone in their support group for assistance.
- All of the examples given here are in tcsh. If you are very familiar with another shell (e.g., bash), and feel comfortable making the necessary alterations to the commands, then feel free to use your other shell. If not, however, we recommend using tcsh.

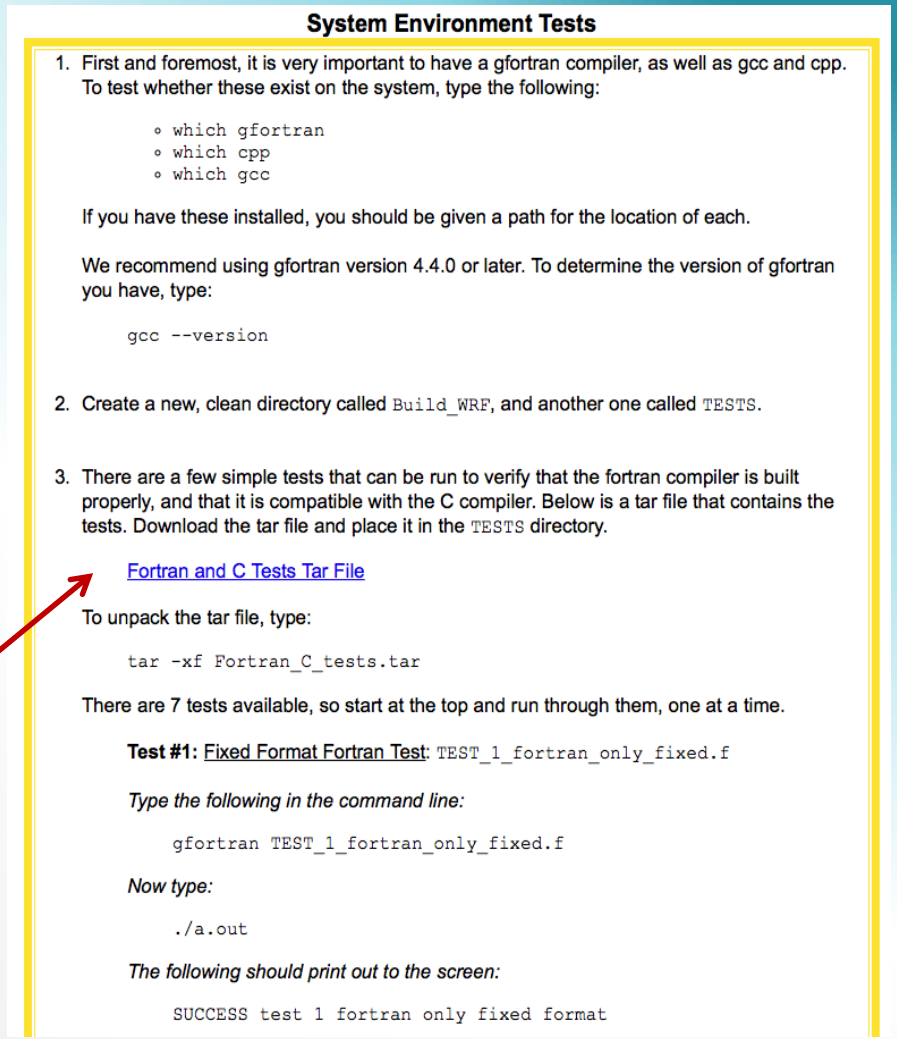
CHECK SYSTEM REQUIREMENTS

- It is mandatory to have a Fortran (e.g., gfortran) compiler, a C compiler, and cpp on your system. To test whether these exist on your system, type:
  - **`which gfortran`**
  - **`which cpp`**
  - **`which gcc`**
  - If installed, you will be given a path for each

- Fortran compiler should be version 4.4.0, or later
  - Check this by typing (csh e.g.):

```
gcc --version
```

- Tests available for checking that your fortran compiler is built properly, and that it is compatible with the C compiler.



System Environment Tests

1. First and foremost, it is very important to have a gfortran compiler, as well as gcc and cpp. To test whether these exist on the system, type the following:
   - `which gfortran`
   - `which cpp`
   - `which gcc`

   If you have these installed, you should be given a path for the location of each.

   We recommend using gfortran version 4.4.0 or later. To determine the version of gfortran you have, type:

   ```
   gcc --version
   ```

2. Create a new, clean directory called `Build_WRF`, and another one called `TESTS`.

3. There are a few simple tests that can be run to verify that the fortran compiler is built properly, and that it is compatible with the C compiler. Below is a tar file that contains the tests. Download the tar file and place it in the `TESTS` directory.

   Fortran and C Tests Tar File

   To unpack the tar file, type:

   ```
   tar -xf Fortran_C_tests.tar
   ```

   There are 7 tests available, so start at the top and run through them, one at a time.

   Test #1: Fixed Format Fortran Test: `TEST_1_fortran_only_fixed.f`

   Type the following in the command line:

   ```
   gfortran TEST_1_fortran_only_fixed.f
   ```

   Now type:

   ```
   ./a.out
   ```

   The following should print out to the screen:

   ```
   SUCCESS test 1 fortran only fixed format
   ```

ADDITIONAL NECESSARY REQUIREMENTS

- Scripting languages (testing available in test package):
  - csh
  - perl
  - sh

- UNIX Commands

ar	awk	head	sed	hostname	sleep
cat	ls	sort	tar	cd	cp
make	touch	mkdir	tr	expr	mv
wc	uname	grep	rm	file	printf
nm	which				

INSTALLING STEPS

- Check system requirements
- ***Installing libraries***
- Download source data
- Compile WRF
- Compile WPS
- Download initial/BC datasets

INSTALLING LIBRARIES

- NetCDF (needed by WRF and WPS)
  - netCDF Version 3 or 4 are acceptable
  - If using netCDF4 capabilities
    http://www2.mmm.ucar.edu/wrf/users/building_netcdf4.html

- Optional libraries for GRIB2 meteorological data support
  - JasPer (JPEG 2000 “lossy” compression library)
  - PNG (“lossless” compression library)
  - Zlib (compression library used by PNG)

- Optional MPI library (for building in parallel):
  - MPICH2

INSTALLING LIBRARIES

- Installation of these libraries (MPICH2, NetCDF, JasPer, zlib, and libpng) is NOT part of the WPS and WRF installation scripts

- **VERY IMPORTANT!**
  - Make sure these libraries are installed using the same compilers as will be used to install WRF and WPS

- Downloads for the libraries, with installation instructions, and library compatibility tests are also included on the compilation website

BEFORE INSTALLING LIBRARIES: SET ENVIRONMENT VARIABLES

```
> setenv DIR directory-where-your-tar-files-are
> setenv CC gcc
> setenv CXX g++
> setenv FC gfortran
> setenv FCFLAGS -m64     # FCFLAGS may be needed on some systems
> setenv F77 gfortran
> setenv FFLAGS -m64      # FFLAGS may be needed on some systems
> setenv LDFLAGS  -L$DIR/grib2/lib
> setenv CPPFLAGS -I$DIR/grib2/include
```

Keep these set until all libraries are built

INSTALLING LIBRARIES: NETCDF

```
> tar xzvf netcdf-4.1.3.tar.gz    # no '.gz' if downloaded to
                                  # most Macs

> cd netcdf-4.1.3

>./configure --prefix=$DIR/netcdf --disable-dap \
--disable-netcdf-4 --disable-shared

> make

> make install

> setenv PATH $DIR/netcdf/bin:$PATH

> setenv NETCDF $DIR/netcdf

> cd ..
```

INSTALLING LIBRARIES: MPICH2

In principle, any implementation of the MPI-2 standard should work with WRF; however, we have the most experience with MPICH

```
> tar xzvf mpich-3.0.4.tar.gz      # no '.gz' if downloaded to
                                   # most Macs
> cd mpich-3.0.4
> ./configure --prefix=$DIR/mpich
> make
> make install
> setenv PATH $DIR/mpich/bin:$PATH
> cd ..
```

INSTALLING LIBRARIES: ZLIB

```
> tar xzvf zlib-1.2.7.tar.gz          # no '.gz' if downloaded to
                                      # most Macs

> cd zlib-1.2.7
>./configure --prefix=$DIR/zlib
> make
> make install
> cd ..
```

INSTALLING LIBRARIES: LIBPNG

```
> tar xzvf libpng-1.2.50.tar.gz   # no '.gz' if downloaded to
                                  # most Macs
> cd libpng-1.2.50
>./configure --prefix=$DIR/libpng
> make
> make install
> cd ..
```

INSTALLING LIBRARIES: JASPER

```
> tar xzvf jasper-1.900.1.tar.gz  # no '.gz' if downloaded to
                                  # most Macs
> cd jasper-1.900.1
> ./configure --prefix=$DIR/jasper
> make
> make install
> cd ..
```

INSTALLING LIBRARIES: COMPATIBILITY

- Make sure libraries are compatible with compilers
- Test 1
  - Fortran + C + netCDF
- Test 2
  - Fortran + C + netCDF + MPI

Library Compatibility Tests

- Once the target machine is able to make small Fortran and C executables (what was verified in the System Environment Tests section), and after the NetCDF and MPI libraries are constructed (two of the libraries from the Building Libraries section), to emulate the WRF code's behavior, two additional small tests are required. We need to verify that the libraries are able to work with the compilers that are to be used for the WPS and WRF builds. Below is a tar file that contans these tests. Download this tar file and place it in the `TESTS` directory:

  Fortran_C_NETCDF_MPI_tests.tar

  To unpack the tar file, type:

  ```
  tar -xf Fortran_C_NETCDF_MPI_tests.tar
  ```

- There are 2 tests:

  1. **Test #1**: Fortran + C + NetCDF

     The NetCDF-only test requires the include file from the NETCDF package be in this directory. Copy the file here:

     ```
     cp ${NETCDF}/include/netcdf.inc .
     ```

     Compile the Fortran and C codes for the purpose of this test (the -c option says to not try to build an executable). Type the following commands:

     ```
     gfortran -c 01_fortran+c+netcdf_f.f
     gcc -c 01_fortran+c+netcdf_c.c
     gfortran 01_fortran+c+netcdf_f.o 01_fortran+c+netcdf_c.o \
          -L${NETCDF}/lib -lnetcdff -lnetcdf
     ./a.out
     ```

     The following should be displayed on your screen:

     ```
     C function called by Fortran
     Values are xx = 2.00 and ii = 1
     SUCCESS test 1 fortran + c + netcdf
     ```

  2. **Test #2**: Fortran + C + NetCDF + MPI

     The NetCDF+MPI test requires include files from both of these packages be in this directory, but the MPI scripts automatically make the `mpif.h` file available without assistance, so no need to copy that one. Copy the NetCDF include file here:

     ```
     cp ${NETCDF}/include/netcdf.inc .
     ```

     Note that the MPI executables `mpif90` and `mpicc` are used below when compiling. Issue the following commands:

     ```
     mpif90 -c 02_fortran+c+netcdf+mpi_f.f
     mpicc -c 02_fortran+c+netcdf+mpi_c.c
     mpif90 02_fortran+c+netcdf+mpi_f.o \
     02_fortran+c+netcdf+mpi_c.o \
          -L${NETCDF}/lib -lnetcdff -lnetcdf
     mpirun ./a.out
     ```

     The following should be displayed on your screen:

     ```
     C function called by Fortran
     Values are xx = 2.00 and ii = 1
     status = 2
     SUCCESS test 2 fortran + c + netcdf + mpi
     ```

INSTALLING STEPS

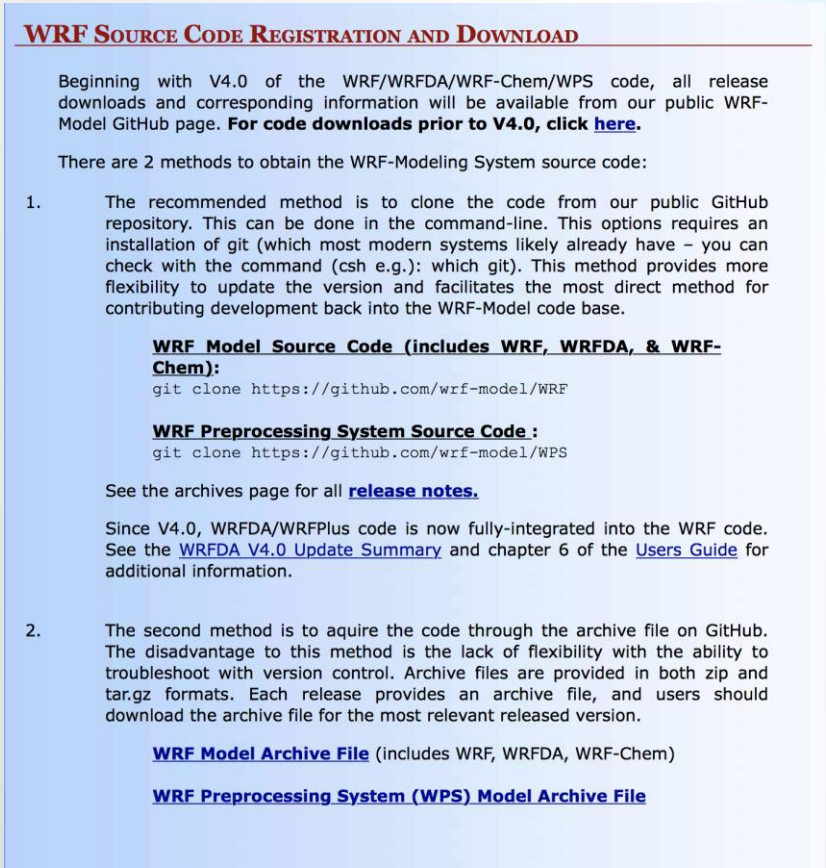
- Check system requirements
- Installing libraries
- ***Download source data***
- Compile WRF
- Compile WPS
- Download initial/BC datasets

DOWNLOAD WRF & WPS CODE

- Download WRF & WPS source code from:

http://www2.mmm.ucar.edu/wrf/users/download/get_source.html

  - Click 'New User,' register and download, or
  - Click 'Returning User,' enter your email, and go to download information page.



WRF Source Code Registration and Download

Beginning with V4.0 of the WRF/WRFDA/WRF-Chem/WPS code, all release downloads and corresponding information will be available from our public WRF-Model GitHub page. **For code downloads prior to V4.0, click here.**

There are 2 methods to obtain the WRF-Modeling System source code:

1. The recommended method is to clone the code from our public GitHub repository. This can be done in the command-line. This options requires an installation of git (which most modern systems likely already have – you can check with the command (csh e.g.): which git). This method provides more flexibility to update the version and facilitates the most direct method for contributing development back into the WRF-Model code base.

   WRF Model Source Code (includes WRF, WRFDA, & WRF-Chem):
   `git clone https://github.com/wrf-model/WRF`

   WRF Preprocessing System Source Code :
   `git clone https://github.com/wrf-model/WPS`

   See the archives page for all **release notes.**

   Since V4.0, WRFDA/WRFPlus code is now fully-integrated into the WRF code. See the WRFDA V4.0 Update Summary and chapter 6 of the Users Guide for additional information.

2. The second method is to aquire the code through the archive file on GitHub. The disadvantage to this method is the lack of flexibility with the ability to troubleshoot with version control. Archive files are provided in both zip and tar.gz formats. Each release provides an archive file, and users should download the archive file for the most relevant released version.

   WRF Model Archive File (includes WRF, WRFDA, WRF-Chem)

   WRF Preprocessing System (WPS) Model Archive File

All Code now available From GitHub!

2 Download Methods:

- Clone from Github
- Download archived tar file from GitHub

DOWNLOAD WRF & WPS CODE

- Cloning WRF from GitHub repository:

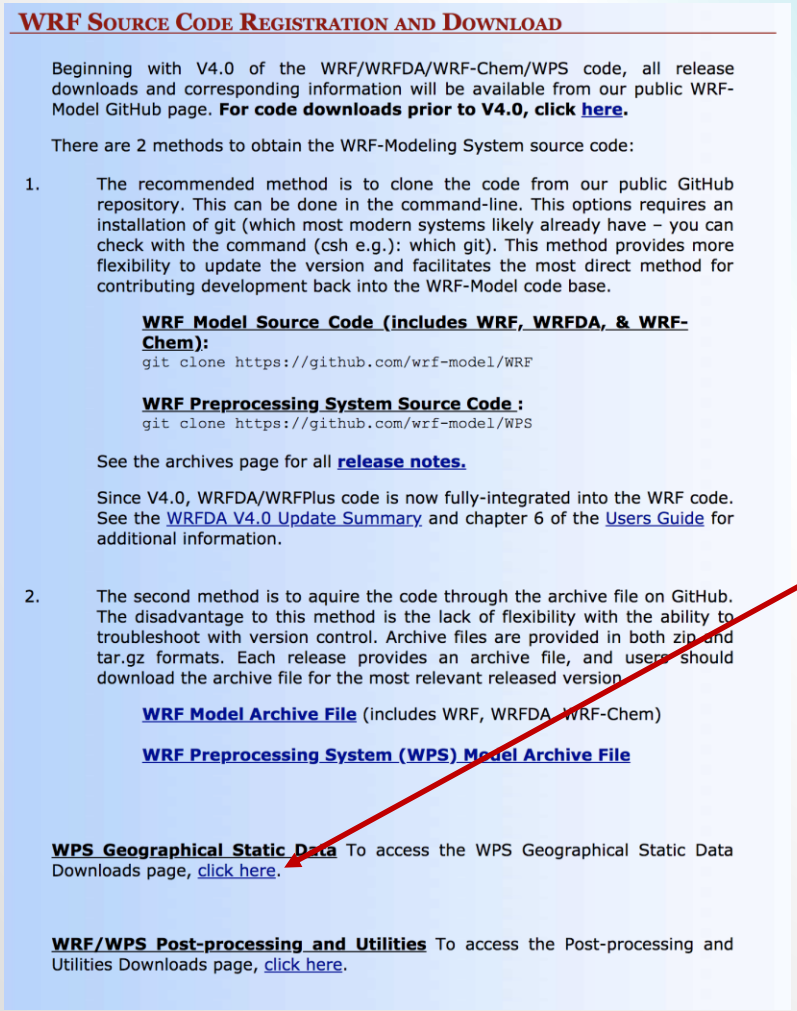
```
Terminal — -tcsh — 146×24
vpn3.ucar.edu:/Users/kkeene/GITHUB>git clone https://github.com/wrf-model/WRF
Cloning into 'WRF'...
remote: Enumerating objects: 77, done.
remote: Counting objects: 100% (77/77), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 56500 (delta 38), reused 29 (delta 21), pack-reused 56423
Receiving objects: 100% (56500/56500), 127.60 MiB | 3.55 MiB/s, done.
Resolving deltas: 100% (43239/43239), done.
Checking out files: 100% (4593/4593), done.
vpn3.ucar.edu:/Users/kkeene/GITHUB>cd WRF
vpn3.ucar.edu:/Users/kkeene/GITHUB/WRF>ls
Makefile  Registry  chem      compile   doc       dyn_exp   external  hydro     main      run       test      var
README    arch      clean     configure dyn_em    dyn_nmm   frame     inc       phys      share     tools     wrftladj
vpn3.ucar.edu:/Users/kkeene/GITHUB/WRF>
```

**Must have 'git' installed on your system!

DOWNLOAD STATIC GEOGRAPHICAL DATA

- From the WRF Download page:

  http://www2.mmm.ucar.edu/wrf/users/download/get_sources_new.php



WRF SOURCE CODE REGISTRATION AND DOWNLOAD

Beginning with V4.0 of the WRF/WRFDA/WRF-Chem/WPS code, all release downloads and corresponding information will be available from our public WRF-Model GitHub page. **For code downloads prior to V4.0, click here.**

There are 2 methods to obtain the WRF-Modeling System source code:

1. The recommended method is to clone the code from our public GitHub repository. This can be done in the command-line. This options requires an installation of git (which most modern systems likely already have – you can check with the command (csh e.g.): which git). This method provides more flexibility to update the version and facilitates the most direct method for contributing development back into the WRF-Model code base.

   WRF Model Source Code (includes WRF, WRFDA, & WRF-Chem):
   `git clone https://github.com/wrf-model/WRF`

   WRF Preprocessing System Source Code :
   `git clone https://github.com/wrf-model/WPS`

   See the archives page for all **release notes.**

   Since V4.0, WRFDA/WRFPlus code is now fully-integrated into the WRF code. See the WRFDA V4.0 Update Summary and chapter 6 of the Users Guide for additional information.

2. The second method is to aquire the code through the archive file on GitHub. The disadvantage to this method is the lack of flexibility with the ability to troubleshoot with version control. Archive files are provided in both zip and tar.gz formats. Each release provides an archive file, and users should download the archive file for the most relevant released version.

   WRF Model Archive File (includes WRF, WRFDA, WRF-Chem)

   WRF Preprocessing System (WPS) Model Archive File

WPS Geographical Static Data To access the WPS Geographical Static Data Downloads page, click here.

WRF/WPS Post-processing and Utilities To access the Post-processing and Utilities Downloads page, click here.

Click Here

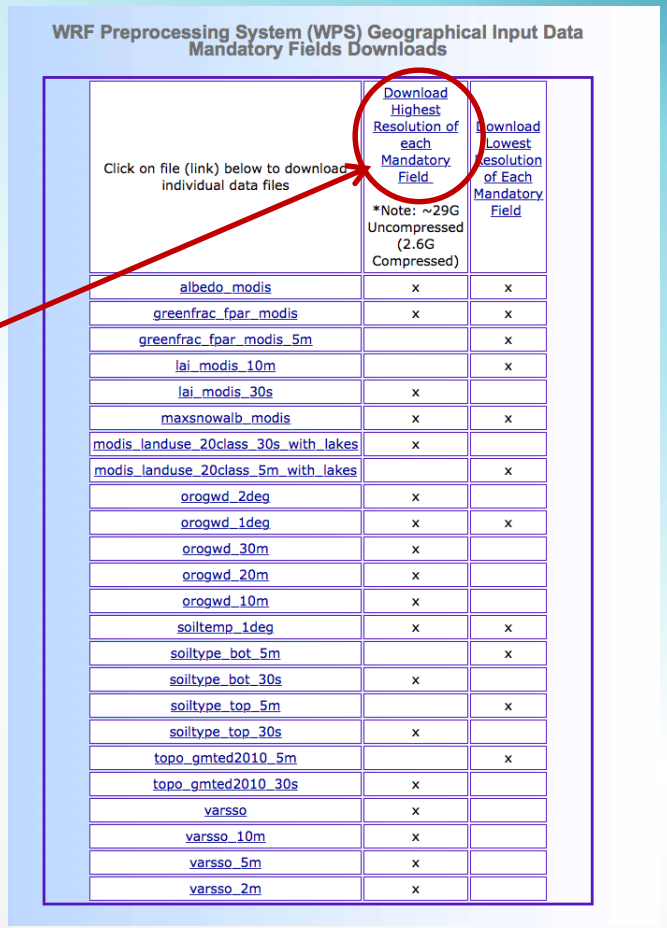
DOWNLOAD STATIC GEOGRAPHICAL DATA

- Geographical Input and Data Download Page:

http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html

geog_high_res_mandatory.tar.gz
~ 29 GB when uncompressed

This is the one you want



WRF Preprocessing System (WPS) Geographical Input Data
Mandatory Fields Downloads

Click on file (link) below to download individual data files	Download Highest Resolution of each Mandatory Field *Note: ~29G Uncompressed (2.6G Compressed)	Download Lowest Resolution of Each Mandatory Field
albedo_modis	x	x
greenfrac_fpar_modis	x	x
greenfrac_fpar_modis_5m		x
lai_modis_10m		x
lai_modis_30s	x	
maxsnowalb_modis	x	x
modis_landuse_20class_30s_with_lakes	x	
modis_landuse_20class_5m_with_lakes		x
orogwd_2deg	x	
orogwd_1deg	x	x
orogwd_30m	x	
orogwd_20m	x	
orogwd_10m	x	
soiltemp_1deg	x	x
soiltype_bot_5m		x
soiltype_bot_30s	x	
soiltype_top_5m		x
soiltype_top_30s	x	
topo_gmted2010_5m		x
topo_gmted2010_30s	x	
varsso	x	
varsso_10m	x	
varsso_5m	x	
varsso_2m	x	

STATIC GEOGRAPHICAL DATA: OTHER OPTIONS

- Geographical Input and Data Download Page:

http://www2.mmm.ucar.edu/wrf/users/download/get_sources_wps_geog.html

WPS Geographical Input Data Mandatory for Specific Applications

Click on file (link) below to download individual data files	Mandatory Use	Combined TAR Files
clayfrac_5m erod sandfrac_5m	Thompson MP Scheme *(mp_physics=28)* and chem	Thompson28 and Chem Tar File
crop groundwater soilgrids	NoahMP LSM *(sf_surface_physics=4)*	NoahMP Tar File
nlcd2011_can_ll_9s nlcd2011_imp_ll_9s nlcd2011_ll_9s	Pleim-Xiu LSM *(sf_surface_physics=7)* **U.S. Only**	Pleim-Xiu Tar File
NUDAPT44_1KM urbfrac_nlcd2011	Urban Physics *(sf_urban_physics=1, 2, or 3)* **U.S. Only**	Urban Tar File
ssib_landuse_10m ssib_landuse_5m	SSiB LSM *(sf_surface_physics=8)*	Urban Tar File
lake_depth	Lake Model *(sf_lake_physics=1)*	
hangl hanis hasynw hasys hasysw hasyw hcnvx hlennw hlens hlensw hlennw hslop hstdv hzmax	NMM Dynamical Core	NMM Static Data Tar File

Optional WPS Geographical Input Data

Click on file (link) below to download individual data files	Optional Use	Combined TAR Files
albedo_ncep greenfrac landuse_30s_with_lakes maxsnowalb	Simulations Older than Year 2000	Older Than 2000 Tar File
bnu_soiltype_bot bnu_soiltype_top	Alternative Data Source for all LSM's	Alternative LSM Data Tar File
modis_landuse_20class_15s	Alternative High-Resolution Data **Does Not Include Lakes**	
nlcd2006_ll_9s	Alternative High-Resolution Data **U.S. Only**	
updated_Iceland_LU	Modified USGS Land cover tiles over Iceland (Includes 'landuse_30s' and 'landuse_30s_with_lakes')	

WRF Weather Research & Forecasting

INSTALLING STEPS

- Check system requirements
- Installing libraries
- Download source data
- ***Compile WRF***
- Compile WPS
- Download initial/BC datasets

CHOOSING A COMPILER

- Compile
  - WRF V4.0
  - dmpar/nesting
  - 4 processors
- Run
  - Single domain
  - Small domain (75x70), 30km resolution
  - 12 hours
  - 8 processors

Compiler	Compile Time	Run Time
GNU 6.3.0 **FREE**	6.82 Mins	3.92 Mins
Intel 17.0.1	46.77 Mins	2.20 Min
PGI 17.9	28.35 Mins	1.95 Min

STEP 1: CONFIGURE FOR WRF

- Inside the WRF/ directory, type: `./configure`

```
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
------------------------------------------------------------------------
Please select from among the following Linux x86_64 options:

  1. (serial)   2. (smpar)   3. (dmpar)   4. (dm+sm)   PGI (pgf90/gcc)
  5. (serial)   6. (smpar)   7. (dmpar)   8. (dm+sm)   PGI (pgf90/pgcc): SGI MPT
  9. (serial)  10. (smpar)  11. (dmpar)  12. (dm+sm)   PGI (pgf90/gcc): PGI accelerator
 13. (serial)  14. (smpar)  15. (dmpar)  16. (dm+sm)   INTEL (ifort/icc)
                                         17. (dm+sm)   INTEL (ifort/icc): Xeon Phi (MIC architecture)
 18. (serial)  19. (smpar)  20. (dmpar)  21. (dm+sm)   INTEL (ifort/icc): Xeon (SNB with AVX mods)
 22. (serial)  23. (smpar)  24. (dmpar)  25. (dm+sm)   INTEL (ifort/icc): SGI MPT
 26. (serial)  27. (smpar)  28. (dmpar)  29. (dm+sm)   INTEL (ifort/icc): IBM POE
 30. (serial)               31. (dmpar)                PATHSCALE (pathf90/pathcc)
 32. (serial)  33. (smpar)  34. (dmpar)  35. (dm+sm)   GNU (gfortran/gcc)
 36. (serial)  37. (smpar)  38. (dmpar)  39. (dm+sm)   IBM (xlf90_r/cc_r)
 40. (serial)  41. (smpar)  42. (dmpar)  43. (dm+sm)   PGI (ftn/gcc): Cray XC CLE
 44. (serial)  45. (smpar)  46. (dmpar)  47. (dm+sm)   CRAY CCE (ftn $(NOOMP)/cc): Cray XE and XC
 48. (serial)  49. (smpar)  50. (dmpar)  51. (dm+sm)   INTEL (ftn/icc): Cray XC
 52. (serial)  53. (smpar)  54. (dmpar)  55. (dm+sm)   PGI (pgf90/pgcc)
 56. (serial)  57. (smpar)  58. (dmpar)  59. (dm+sm)   PGI (pgf90/gcc): -f90=pgf90
 60. (serial)  61. (smpar)  62. (dmpar)  63. (dm+sm)   PGI (pgf90/pgcc): -f90=pgf90
 64. (serial)  65. (smpar)  66. (dmpar)  67. (dm+sm)   INTEL (ifort/icc): HSW/BDW
 68. (serial)  69. (smpar)  70. (dmpar)  71. (dm+sm)   INTEL (ifort/icc): KNL MIC
 72. (serial)  73. (smpar)  74. (dmpar)  75. (dm+sm)   FUJITSU (frtpx/fccpx): FX10/FX100 SPARC64 IXfx/Xlfx

Enter selection [1-75] : 34
------------------------------------------------------------------------
Compile for nesting? (1=basic, 2=preset moves, 3=vortex following) [default 1]:
```

CONFIGURE OPTIONS FOR WRF

DEBUGGING OPTIONS

- `./configure -d`
  - No optimization
  - Extra debugging
- `./configure -D`
  - No optimization
  - Checks uninitialized variables, floating point traps, etc.
- `./configure -r8`
  - Double-precision
    - Works for GNU, Intel, & PGI compilers

OLDER VERSIONS

- **Large-file support**
  - For output files > 2GB
  - Default since V3.9
  - Before configuring, set (csh e.g.)

```
setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1
```

- **Hybrid coordinate**
  - Default since V4.0
  - Version 3.9 – v3.9.1.1 needs:
  - `./configure -hyb`

PARALLEL COMPILE OPTION FOR WRF

- To build WRF with multiple compilers, set (csh e.g.):

```
setenv J "-j2"
```

# of Processors	Time to Compiler
1	17.25 Mins
2	9.95 Mins
3	8.05 Mins
4	6.82 Mins
5	6.32 Mins
6	6.12 Mins

Compiled with GNU V6.3.0

STEP 2: COMPILE WRF

- In the WRF/ directory, type:

```
./compile em_case >& log.compile
```

Where **em_case** is one of the following
(type **./compile** to see all options)

em_real (3d real case)

em_quarter_ss
em_b_wave
em_les
em_heldsuarez
em_tropical_cyclone
em_convrad

3d Ideal

em_hill2d_x
em_squall2d_x
em_squall2d_y
em_grav2d_x
em_seabreeze2d_x

2d Ideal

em_scm_xy (1d ideal)

Compilation time depends on compiler

SUCCESSFUL COMPILATION

- If the compilation is successful, you should find these executables in **`WRF/main`** (non-zero size):

Real data case:

`wrf.exe` – model executable

`real.exe` – real data initialization

`ndown.exe` – one-way nesting

`tc.exe` – for tc bogusing (serial only)

Ideal case:

`wrf.exe` – model executable

`ideal.exe` – ideal case initialization

***Note: Each ideal case compile creates a different executable, but with the same name**

- These executables are linked to 2 different directories (**`WRF/run`** and **`WRF/test/em_real`**). You can go to either place to run WRF.

UNSUCCESSFUL COMPILATION

- Use your 'log.compile' file to search for compiler errors!
  - Search for **`Error`** with a capital **`E`**
- Use our Frequently Asked Questions web page for help
  - www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wrf_intallation.html
- Visit the wrfhelp Forum:
  http://forum.mmm.ucar.edu/
- Before recompiling:
  - issue a '**`clean -a`**'
  - Reconfigure: If you need to make changes to the configure.wrf file, do this after issuing ./configure, and then save the edited file.
  - Recompile

INSTALLING STEPS

- Check system requirements
- Installing libraries
- Download source data
- Compile WRF
- ***Compile WPS***
- Download initial/BC datasets

INSTALLING STEPS

- Check system requirements
- Installing libraries
- Download source data
- Compile WRF
- ***Compile WPS*** ← **MUST BE BUILT AFTER WRF**
- Download initial/BC datasets

STEP 1: CONFIGURE FOR WPS

- Inside the WPS/ directory, type:

  ./configure

```
$JASPERLIB or $JASPERINC not found in environment. Using default values for library paths...
-----------------------------------------------------------------------
Please select from among the following supported platforms.

   1.  Linux x86_64, gfortran    (serial)
   2.  Linux x86_64, gfortran    (serial_NO_GRIB2)
   3.  Linux x86_64, gfortran    (dmpar)
   4.  Linux x86_64, gfortran    (dmpar_NO_GRIB2)
   5.  Linux x86_64, PGI compiler   (serial)
   6.  Linux x86_64, PGI compiler   (serial_NO_GRIB2)
   7.  Linux x86_64, PGI compiler   (dmpar)
   8.  Linux x86_64, PGI compiler   (dmpar_NO_GRIB2)
   9.  Linux x86_64, PGI compiler, SGI MPT   (serial)
  10.  Linux x86_64, PGI compiler, SGI MPT   (serial_NO_GRIB2)
  11.  Linux x86_64, PGI compiler, SGI MPT   (dmpar)
  12.  Linux x86_64, PGI compiler, SGI MPT   (dmpar_NO_GRIB2)
```

- Choose to compile WPS **serially**, even if you compile WRF with a parallel option (unless you have a very large domain)

  **NOTE: if you do compile WPS in parallel, ungrib.exe must run serially

- Output from configuration: a file called 'configure.wps'

STEP 2: COMPILE WPS

- In the WPS/ directory, type:

```
./compile >& log.compile
```

- Compilation should only take a few minutes

- If successful, these executables should be in your WPS/ directory (and they are linked, respectively, from their source code directories):

```
geogrid.exe -> geogrid/src/geogrid.exe
ungrib.exe -> ungrib/src/ungrib.exe
metgrid.exe -> metgrid/src/metgrid.exe
```

UNSUCCESSFUL WPS COMPILATION

No `geogrid.exe` or `metgrid.exe`

- WPS makes use of the external I/O libraries in the *`WRF/external/`* directory - The libraries are built when WRF is installed
- Check that you used the same compiler (and version) as you used to compile WRF
- Check that you are using the same netCDF that you used to build WRF
- Have you changed the name or path of the WRF/ directory?
  - If so, you need to change the following line in the configure.wps file:

    `WRF_DIR = ../WRF`
  - Save the file and recompile

UNSUCCESSFUL WPS COMPILATION

No `ungrib.exe`

- Make sure you have installed your jasper, zlib, and libpng libraries correctly.

- Make sure that you are using the correct path and format for the following lines in the configure.wps file

```
COMPRESSION_LIBS = -L/${DIR}/UNGRIB_LIBRARIES/lib -ljasper -lpng -lz
COMPRESSION_INC = -I/${DIR}/UNGRIB_LIBRARIES/include
```

Save configure.wps and recompile

THE CLEAN COMMAND

- The ‘./clean –a’ command is something that should be used when you have made corrections to your configure.wrf file, configure.wps file, or any changes to the registry. If you have made any of these changes, or if you plan to recompile your code from scratch, you must issue a ‘clean –a’ before recompiling.

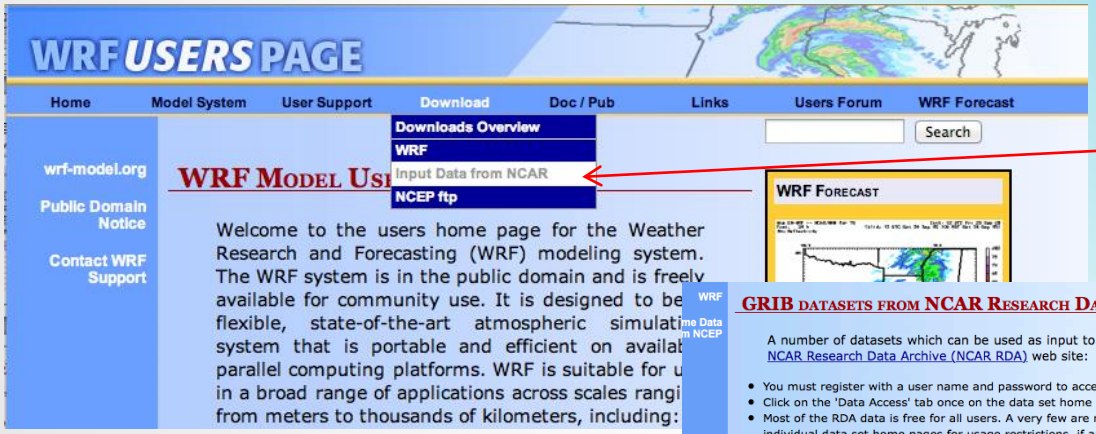
- If you make any changes to any subroutines within the code, you will need to recompile your code, but you do NOT need to issue the ‘./clean –a’ command, nor do you need to reconfigure. You will simply just recompile. This compilation should take a lot less time than a clean compile.

INSTALLING STEPS

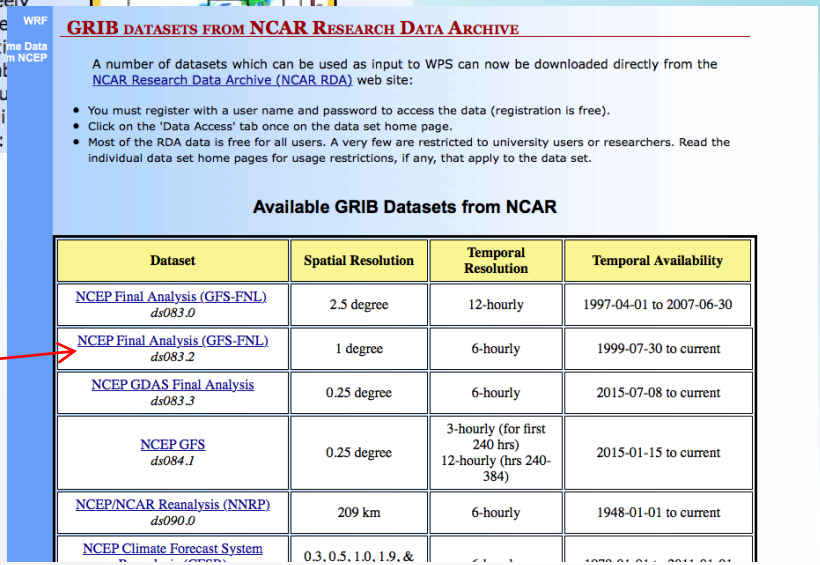
- Check system requirements
- Installing libraries
- Download source data
- Compile WRF
- Compile WPS
- ***Download initial/BC datasets***

DOWNLOAD DATASETS

- From the WRF Users' page: **http://www2.mmm.ucar.edu/wrf/users/**



Step 1: Click Download, then scroll down and click 'Input Data from NCAR'



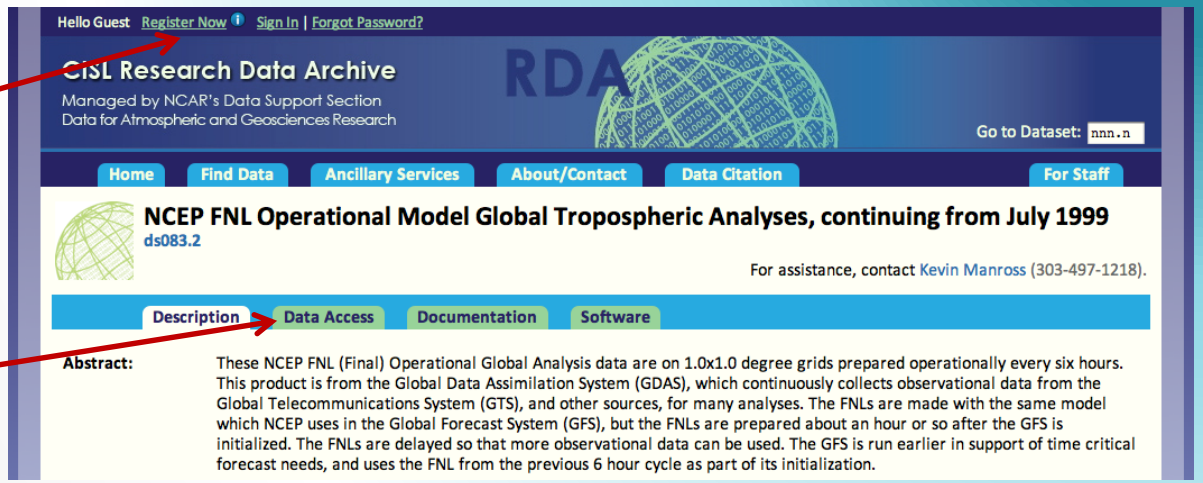
Step 2: Click the dataset you wish to use (for this example, we will use 'FNL from GFS')

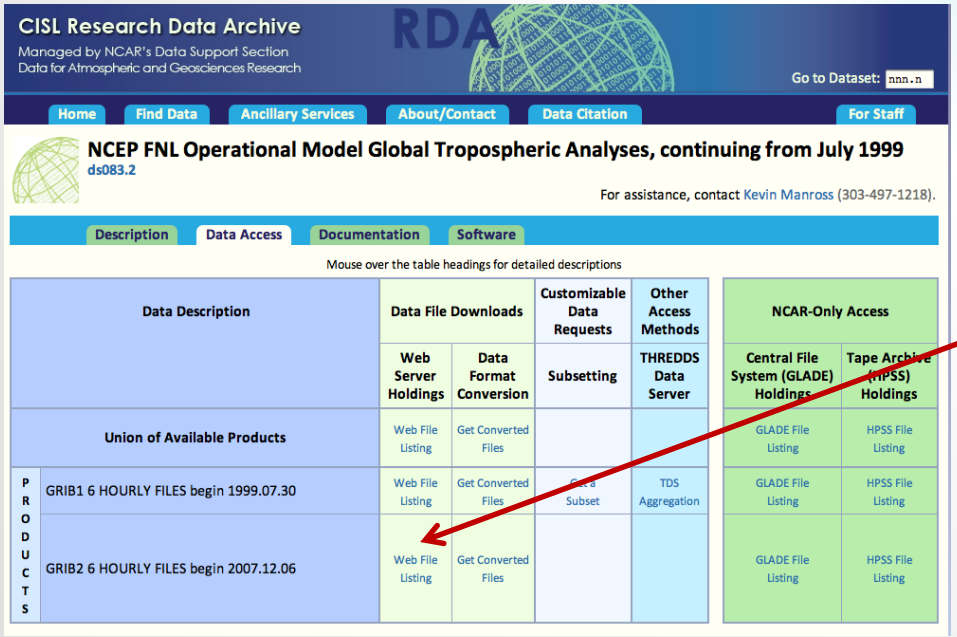
*Note: The NOMADS site has several types of useful data: http://nomads.ncdc.noaa.gov

DOWNLOAD DATASETS (CONT'D)

Step 3: Register, or sign in, if you already have an account

Step 4: Click 'Data Access'

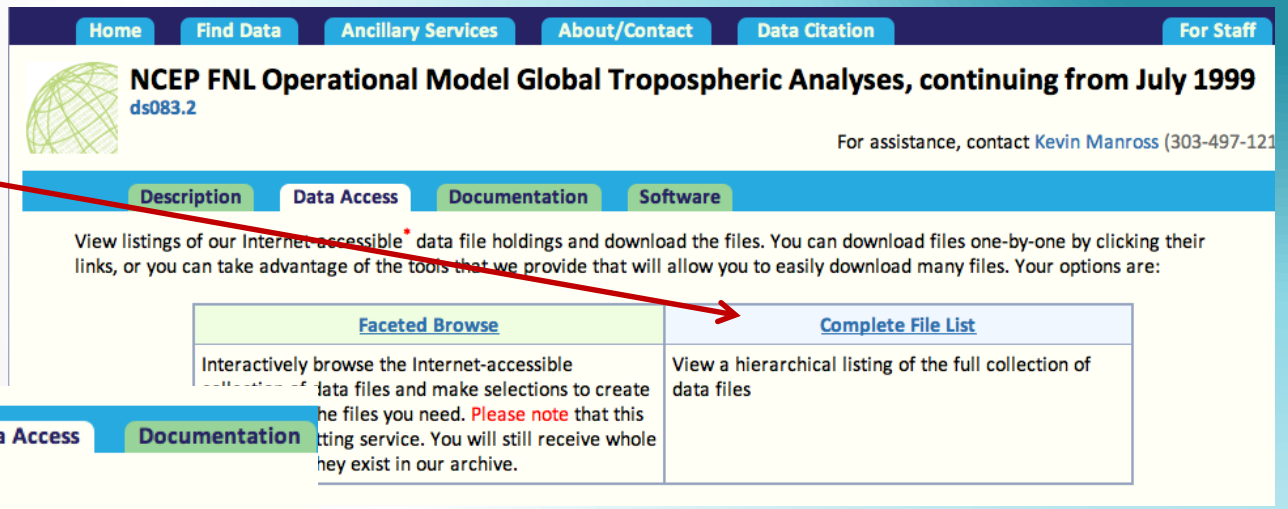


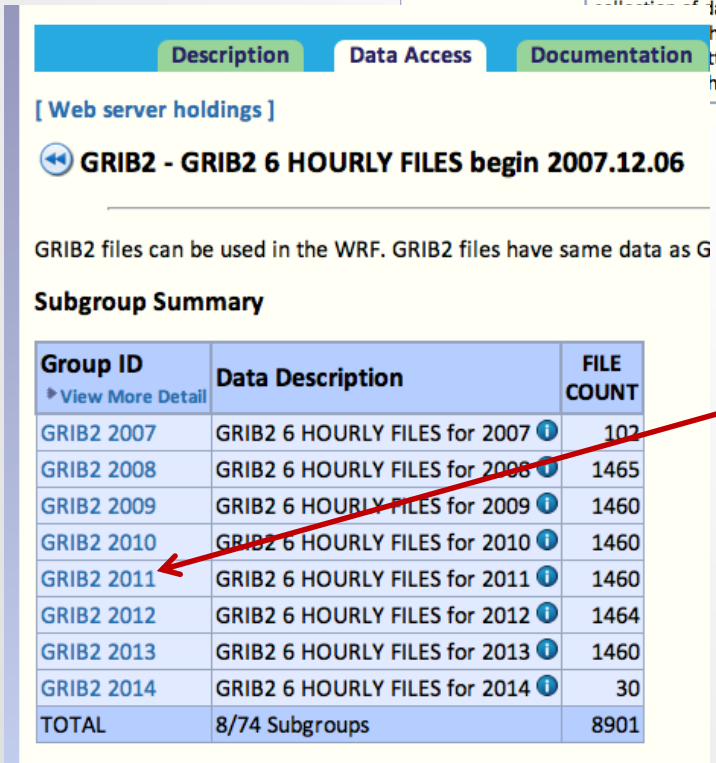


Step 5: Click 'Web File Listing' for the span of years you need

DOWNLOAD DATASETS (CONT'D)

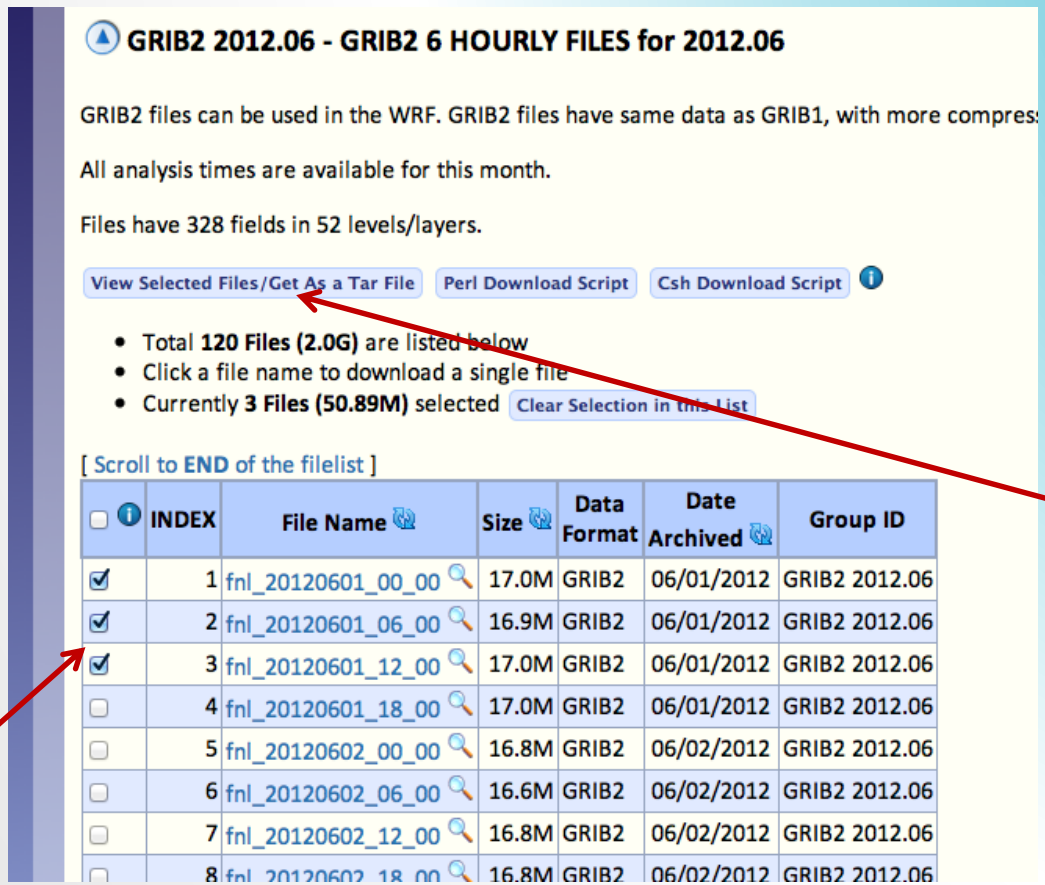
Step 6: Click 'Complete File List'





Step 7: Click the year you need. After this, You will click the month you need (not shown)

DOWNLOAD DATASETS (CONT'D)



GRIB2 2012.06 - GRIB2 6 HOURLY FILES for 2012.06

GRIB2 files can be used in the WRF. GRIB2 files have same data as GRIB1, with more compres

All analysis times are available for this month.

Files have 328 fields in 52 levels/layers.

View Selected Files/Get As a Tar File | Perl Download Script | Csh Download Script

- Total **120 Files (2.0G)** are listed below
- Click a file name to download a single file
- Currently **3 Files (50.89M)** selected Clear Selection in this List

[Scroll to **END** of the filelist]

	INDEX	File Name	Size	Data Format	Date Archived	Group ID
☑	1	fnl_20120601_00_00	17.0M	GRIB2	06/01/2012	GRIB2 2012.06
☑	2	fnl_20120601_06_00	16.9M	GRIB2	06/01/2012	GRIB2 2012.06
☑	3	fnl_20120601_12_00	17.0M	GRIB2	06/01/2012	GRIB2 2012.06
☐	4	fnl_20120601_18_00	17.0M	GRIB2	06/01/2012	GRIB2 2012.06
☐	5	fnl_20120602_00_00	16.8M	GRIB2	06/02/2012	GRIB2 2012.06
☐	6	fnl_20120602_06_00	16.6M	GRIB2	06/02/2012	GRIB2 2012.06
☐	7	fnl_20120602_12_00	16.8M	GRIB2	06/02/2012	GRIB2 2012.06
☐	8	fnl_20120602_18_00	16.8M	GRIB2	06/02/2012	GRIB2 2012.06

Step 8: Click a box for each time span that you need

Step 9: Once you have chosen All your times, click on the 'View Selected Files/Get As a Tar File' button To download one tar file with all your Dates/times

STEPS TO RUN WRF

- Run WPS processes from the *WPS/* directory
- cd to *WRF/run* or *WRF/test/em_real*
- Link metgrid output files from *WPS/* to running directory
- Make appropriate edits to namelist.input file
- Run real.exe: *mpirun –np XX ./real.exe*
- Run wrf.exe: *mpirun –np XX ./wrf.exe*

QUESTIONS?