WRF & WPS: COMPILATION PROCESS

Kelly Werner
NCAR/MMM
January 2020

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

INSTALLING STEPS

- ***Check system requirements***
- Installing libraries
- Obtain source code
- Compile WRF
- Compile WPS
- Download static geographical data

SYSTEM REQUIREMENTS

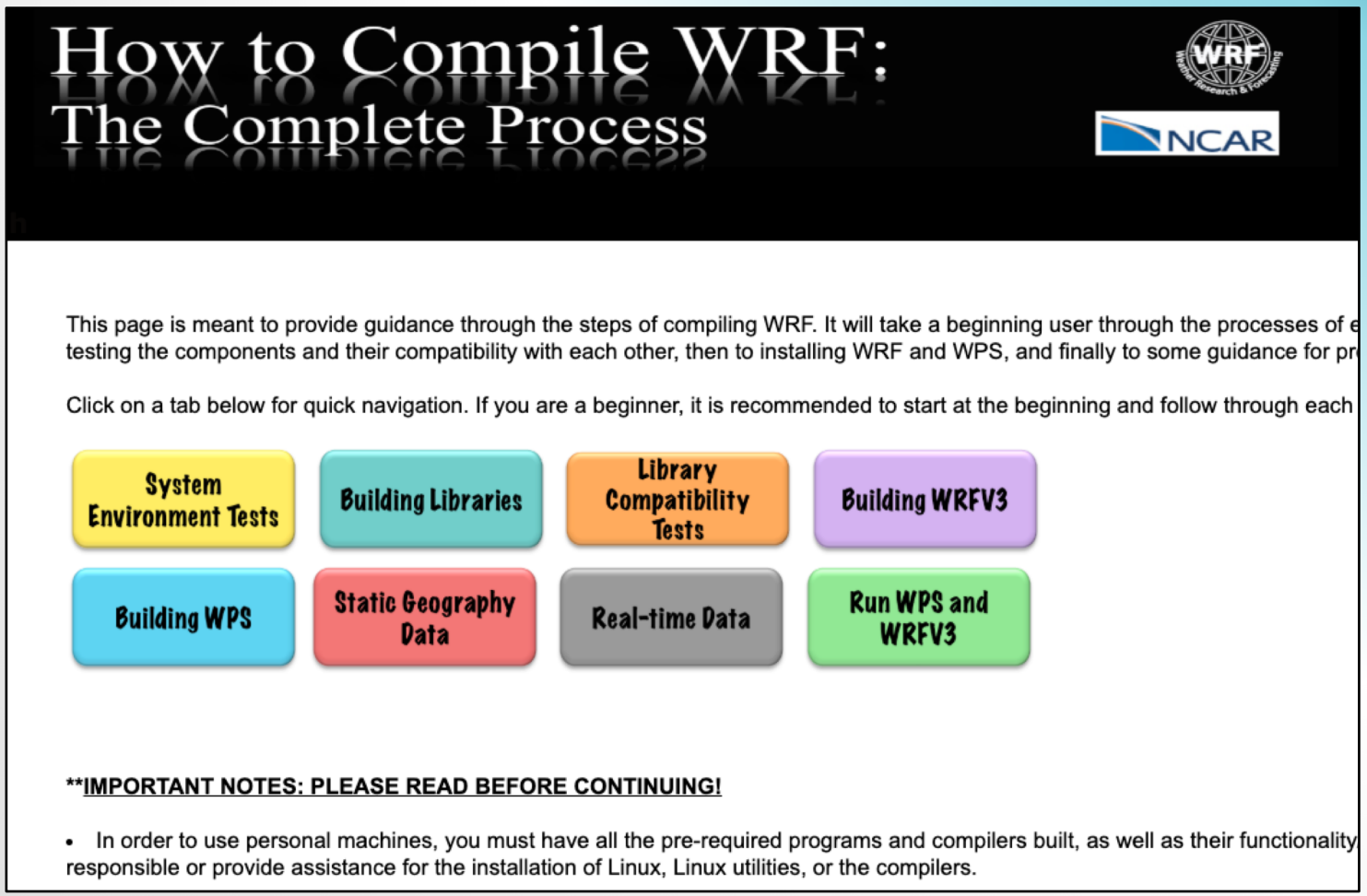
- On what kinds of systems will WRF run?
  - Generally any 32- or 64-bit hardware, running a UNIX-like operating system
  - You may also use dual-booting into a UNIX-like OS (e.g., Windows with Linux built parallel)

- Examples of acceptable systems:
  - Laptops, desktops, and clusters running Linux
  - Laptops and desktops running MacOS
  - 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 e
testing the components and their compatibility with each other, then to installing WRF and WPS, and finally to some guidance for pr

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

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
responsible or provide assistance for the installation of Linux, Linux utilities, or the compilers.

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 a version that supports Fortran2003 standard (at least v4.6)
  - 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 a Fortran compiler that supports Fortran2003 standard (version 4.6 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.
   NOTE: If any of these tests fail, you will need to contact the systems administrator at your institution for help, as these are specific to your particular environment, and we do not have the resources to support these types of errors.

   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
   ```

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***
- Obtain source code
- Compile WRF
- Compile WPS
- Download static geographical data

INSTALLING LIBRARIES

- NetCDF (needed by WRF and WPS)
  - netCDF Version 3 or 4
  - 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):
  - E.g., MPICH2 or OpenMPI

INSTALLING LIBRARIES

- These libraries (MPICH2, NetCDF, JasPer, zlib, & libpng) are NOT included in the WPS & WRF installation packages

- Compilation website includes library files for download, and includes
  - Installation instructions
  - Library compatibility tests

- **VERY IMPORTANT!**

  Make sure libraries are installed using the same compilers as will be used to install WRF & WPS

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 JASPERLIB $DIR/grib2/lib
> setenv JASPERINC $DIR/grib2/include
> setenv LDFLAGS -L$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 ..
```

SET PATHS IN ENVIRONMENT SCRIPT

- E.g., .cshrc, .bash, .tcshrc

```
> setenv DIR directory-where-your-tar-files-are
> setenv PATH $DIR/mpich/bin $DIR/netcdf/bin:$PATH
> setenv LD_LIBRARY_PATH $DIR/grib2/lib
> setenv JASPERLIB $DIR/grib2/lib
> setenv JASPERINC $DIR/grib2/include
```

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.
  NOTE: If any of these tests fail, you will need to contact the systems administrator at your institution for help, as these are specific to your particular environment, and we do not have the resources to support these types of errors.

  Below is a tar file that contans these tests. Download this tar file and place it in the `TESTS` directory, and then "cd" into 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 .
     ```

INSTALLING STEPS

- Check system requirements
- Installing libraries
- ***Obtain source code***
- Compile WRF
- Compile WPS
- Download static geographical data

OBTAIN WRF & WPS CODE

- WRF & WPS source code from:

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

- Click 'New User,' and then register, or
- Click 'Returning User,' enter your email, and go to the 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

Code available from GitHub!

2 Methods to obtain code:

- Clone from Github
- Download archived tar file from GitHub

OBTAIN 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!

INSTALLING STEPS

- Check system requirements
- Installing libraries
- Obtain source code
- ***Compile WRF***
- Compile WPS
- Download static geographical data

CHOOSING A COMPILER

- Compile
  - WRF V4.0
  - dmpar/nesting
  - 4 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

- Run
  - Single domain
  - Small domain (75x70), 30km resolution
  - 12 hours
  - 8 processors

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]:
```

- Configuration output: `configure.wrf`

WRF CONFIGURATION 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*

PARALLEL COMPILE OPTION FOR WRF

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

  `setenv J "-j 2"`

  Before or after configure

# 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 should take ~30 mins

SUCCESSFUL COMPILATION

- If the compilation is successful, you should see 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 (can only be run serially)

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 errors!
  - Search for 'Error' with a capital 'E'
- Use our Frequently Asked Questions forum page for help
- 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

./CLEAN -A

- The ‘./clean –a’ command should be used when modifications have been made to the configure.wrf(wps) file, or any changes to the registry. If so, issue ‘clean –a’ prior to recompiling.

- Modifications to subroutines within the code will require a recompile, but **DO NOT** require a ‘clean –a’, nor a reconfigure. Simply recompile. This compilation should be much faster than a clean compile.

INSTALLING STEPS

- Check system requirements
- Installing libraries
- Obtain source code
- Compile WRF
- ***Compile WPS***
- Download static geographical data

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)
```

- Always choose a **serial** compile for WPS (even if you compile WRF with a parallel option)
  - Exception: You are using a VERY large domain (1000's x 1000's)

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

- Configuration output: `configure.wps`

STEP 2: COMPILE WPS

- In the WPS/ directory, type:

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

- Compilation should be quick.

- If successful, these executables should be in your WPS/ directory (linked 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`

- Make sure WRF compiled successfully.
  - 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 are using the same compiler (and version) as used to compile WRF.
- Check that you are using the same netCDF (and version) as 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`
  - Beginning V4.0: set WRF_DIR environment variable (prior to configure):

    `setenv WRF_DIR` *`path_to_WRF/`*`WRF`
  - Save the configure file and recompile

UNSUCCESSFUL WPS COMPILATION

No `ungrib.exe`

- Make sure jasper, zlib, and libpng libraries are correctly installed.
- 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

INSTALLING STEPS

- Check system requirements
- Installing libraries
- Obtain source code
- Compile WRF
- Compile WPS
- ***Download static geographical data***

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.

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	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	

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	Thompson MP Scheme *(mp_physics=28)* and chem	Thompson28 and Chem Tar File
erod		
sandfrac_5m		
crop	NoahMP LSM *(sf_surface_physics=4)*	NoahMP Tar File
groundwater		
soilgrids		
nlcd2011_can_ll_9s	Pleim-Xiu LSM *(sf_surface_physics=7)* **U.S. Only**	Pleim-Xiu Tar File
nlcd2011_imp_ll_9s		
nlcd2011_ll_9s		
NUDAPT44_1KM	Urban Physics *(sf_urban_physics=1, 2,*	

Optional WPS Geographical Input Data

Click on file (link) below to download individual data files	Optional Use	Combined TAR Files
albedo_ncep	Simulations Older than Year 2000	Older Than 2000 Tar File
greenfrac		
landuse_30s_with_lakes		
maxsnowalb		
bnu_soiltype_bot	Alternative Data Source for all LSM's	Alternative LSM Data Tar File
bnu_soiltype_top		
modis_landuse_20class_15s	Alternative High-resolution Data	
modis_landuse_20class_15s_with_lakes		
	Alternative High-	

QUESTIONS?