WRF & WPS: Compile

Kelly Werner

NCAR/MMM

January 2017

Installing Steps

- *Check system requirements*
- Installing libraries
- Download source data
- Compile WRFV3
- 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

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.

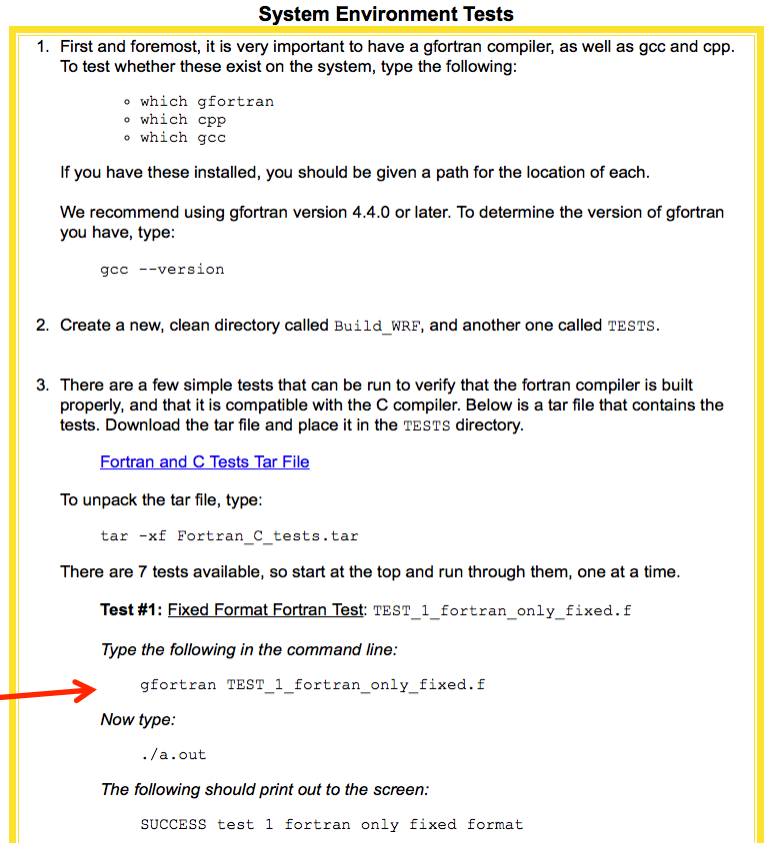
Check System Requirements

- 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

  (for csh):

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

Installing Steps

- Check system requirements
- *Installing libraries*
- Download source data
- Compile WRFV3
- Compile WPS
- Download initial/BC datasets

Installing Libraries

- NetCDF (needed by WRF and WPS)
  - netCDF Version 3 or 4 are acceptable
- 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

Installing Libraries: NetCDF

```
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

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

- *Assuming environment variables for netCDF install are already set:*

```
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

- *Assuming environment variables from netCDF install are already set:*

```
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

- *Assuming environment variables from netCDF install are already set*

```
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

- *Assuming environment variables from netCDF install are already set*

```
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 WRFV3
- Compile WPS
- Download initial/BC datasets

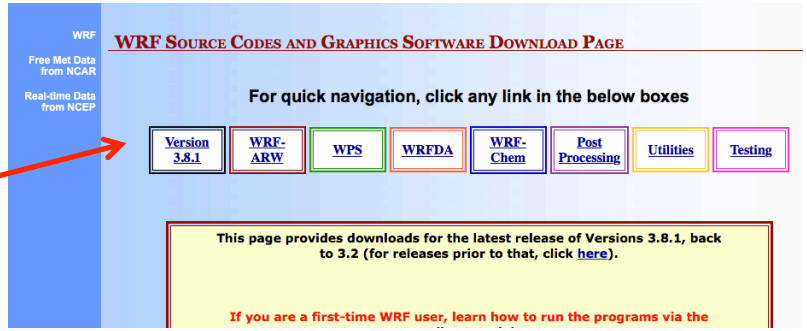
Download WRF & WPS Code

- Download WRF & WPS source code from:

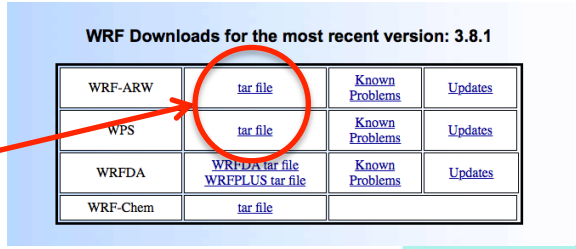
  http://www2.mmm.ucar.edu/wrf/users/download/get_source.html

  – Click 'New User,' register and download, or
  – Click 'Returning User,' enter your email, and download

Step 1:
Click here for the latest released code (recommended)

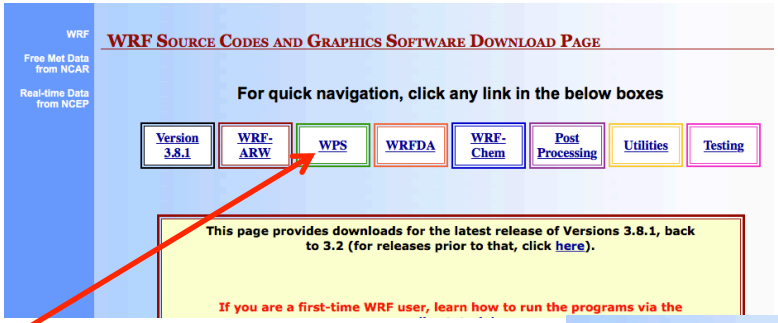


Step 2:
Click on tar files to download



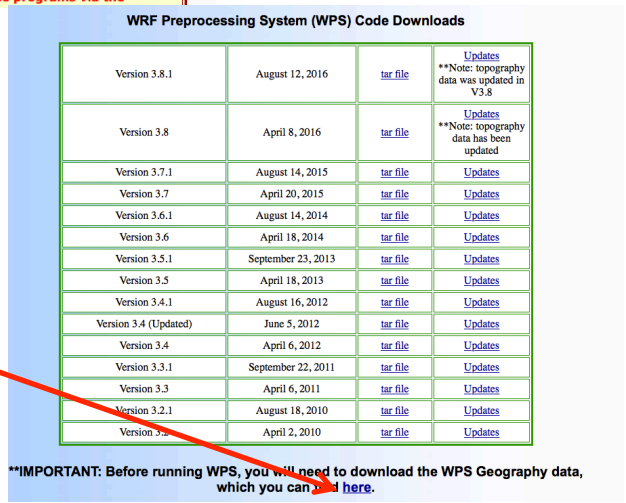
Download Static Geographical Data

- From the WRF Download page: http://www2.mmm.ucar.edu/wrf/users/download/get_sources.html



Step 1: Click 'WPS' box

Step 2: Click 'here' to get geography data



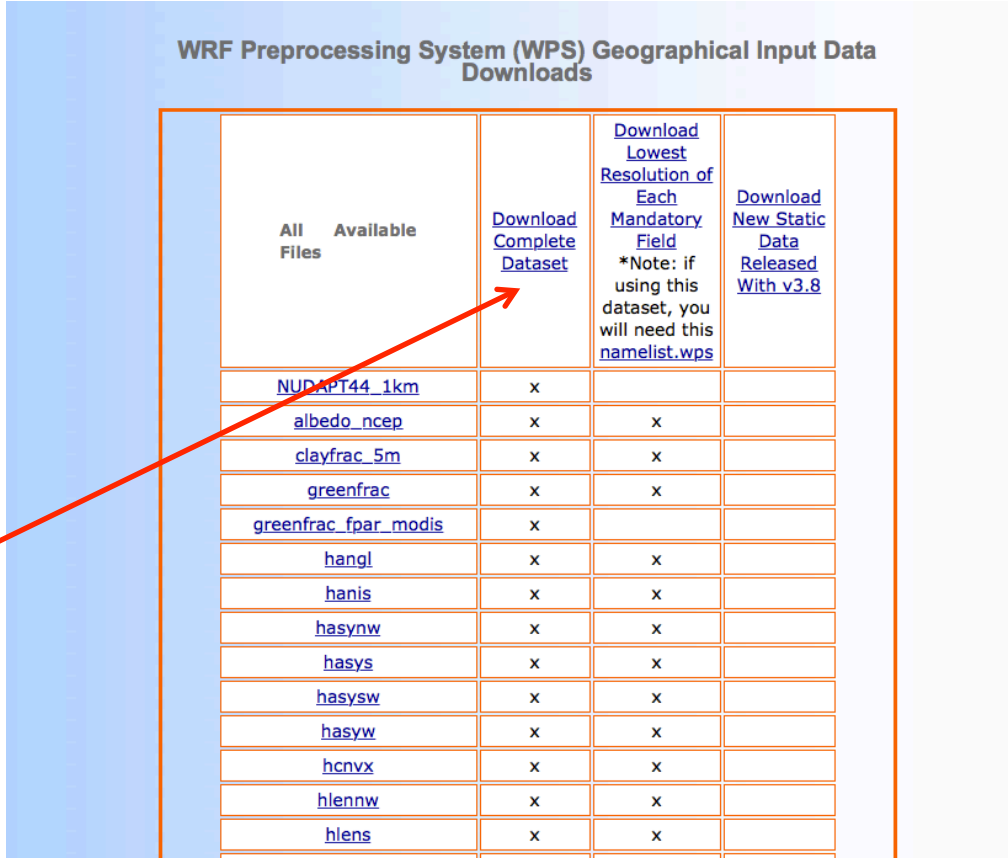
Download Static Geographical Data

- Geographical Input and Data Download Page:

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

geog.tar.gz
~ 15 GB when uncompressed

This is the one you want



WRF Preprocessing System (WPS) Geographical Input Data Downloads

All Available Files	Download Complete Dataset	Download Lowest Resolution of Each Mandatory Field *Note: if using this dataset, you will need this namelist.wps	Download New Static Data Released With v3.8
NUDAPT44_1km	x		
albedo_ncep	x	x	
clayfrac_5m	x	x	
greenfrac	x	x	
greenfrac_fpar_modis	x		
hangl	x	x	
hanis	x	x	
hasynw	x	x	
hasys	x	x	
hasysw	x	x	
hasyw	x	x	
hcnvx	x	x	
hlennw	x	x	
hlens	x	x	

Installing Steps

- Check system requirements
- Installing libraries
- Download source data
- *Compile WRFV3*
- Compile WPS
- Download initial/BC datasets

Choosing a Compiler

Compiler	Compile Time	Run Time
GNU 4.8.2 **FREE**	12.63 Mins	4.18 Mins
Intel 12.1.5	27.75 Mins	3.88 Mins
PGI 13.3-0	24.86 Mins	4.25 Mins

*Compile: dmpar/nesting, no large-file support

*Run: single domain, small domain (74x61), 6 hours, 16 processors

Step 1: Configure for WRFV3

- Inside the WRFV3/ directory, type: **`./configure`**

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /glade/apps/opt/netcdf/4.3.0/intel/12.1.5
PHDF5 not set in environment. Will configure WRF for use without.
Will use 'time' to report timing information
$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/gcc): 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

Enter selection [1-63] :


Compile for nesting? (0=no nesting, 1=basic, 2=preset moves, 3=vortex following) [default 0]:
```

- Output from configuration: a file called ‘configure.wrf’

Configure Options for WRFV3

Debugging Options

- `./configure -d`
  - No optimization
  - Extra debugging
- `./configure -D`
  - No optimization
  - Checks uninitialized variables, floating point traps, etc.
  - Useful for adding/updating new code
- `./configure -r8`
  - Double precision for Intel, GNU, and PGI

Large File Support

- `setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1`
  - > 2GB
  - Before configuring
  - Built-in since V3.9

Parallel Compile Option for WRFV3

- To build WRF in parallel
  - *setenv J "-j 2"*

# of Processors	Time to Compiler
1	22.8 Mins
2	14.92 Mins
3	9.33 Mins
4	8.02 Mins
5	7.23 Mins
6	6.68 Mins

*Around 4 processors, it reaches state of equilibrium

* This test done with GNU compiler

configure.wrf File: Useful Tips

- NETCDFPATH : internally set by build system based on $NETCDF

- PNETCDF = For users who have access to parallel netcdf, use the environment variable PNETCDF identically to how NETCDF is set (point to the PNETCDF top-level directory)

Step 2: Compile WRFV3

- In the WRFV3/ directory, type:

  `./compile em_case >& log.compile`

Important in case there are compile problems

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

`em_real` (3d real case)

3d Ideal:
- **`em_quarter_ss`**
- **`em_b_wave`**
- **`em_les`**
- **`em_heldsuarez`**
- **`em_tropical_cyclone`**
- **`em_convrad`**

2d Ideal:
- **`em_hill2d_x`**
- **`em_squall2d_x`**
- **`em_squall2d_y`**
- **`em_grav2d_x`**
- **`em_seabreeze2d_x`**

`em_scm_xy` (1d ideal)

Compilation should take ~30 mins

Successful Compilation

- If the compilation is successful, you should find these executables in **`WRFV3/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 (**`WRFV3/run`** and **`WRFV3/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 web page for help
  - www2.mmm.ucar.edu/wrf/users/FAQ_files/FAQ_wrf_intallation.html
- 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
- Contact wrfhelp@ucar.edu

Installing Steps

- Check system requirements
- Installing libraries
- Download source data
- Compile WRFV3
- *Compile WPS*
- Download initial/BC datasets

Step 1: Configure for WPS

- Inside the WPS/ directory, type: **`./configure`**

```
Will use NETCDF in dir: /glade/apps/opt/netcdf/4.3.0/intel/12.1.5
$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)
```

- Choose to compile WPS **serially**, even if you compile WRFV3 in parallel (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 *WRFV3/external/* directory - The libraries are built when WRF is installed
- Check that you used the exact same compiler (and version) as you used to compile WRFV3
- Check that you are using the same netCDF that you used to build WRFV3
- Have you changed the name or path of the WRFV3/ directory?
  - If so, you need to change the following line in the configure.wps file:
    WRF_DIR = ../WRFV3
  - 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

./clean -a

- 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.

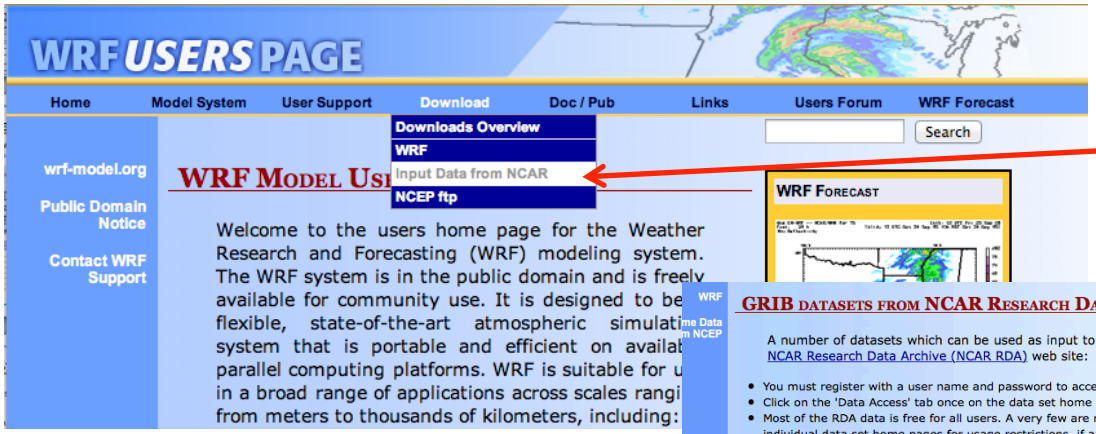
- If you made any changes to any subroutines within the code, you will need to recompile your code, but you do NOT need to issue the ‘clean –a’ command, nor do you need to reconfigure. You will simply just recompile. This compilation should take a lot less time than a clean compile.

Installing Steps

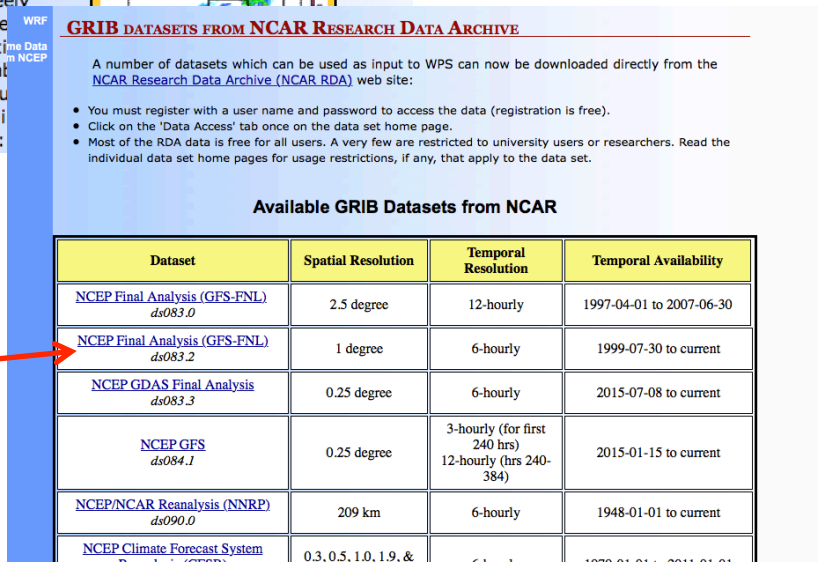
- Check system requirements
- Installing libraries
- Download source data
- Compile WRFV3
- Compile WPS
- *Download initial/BC datasets*

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'



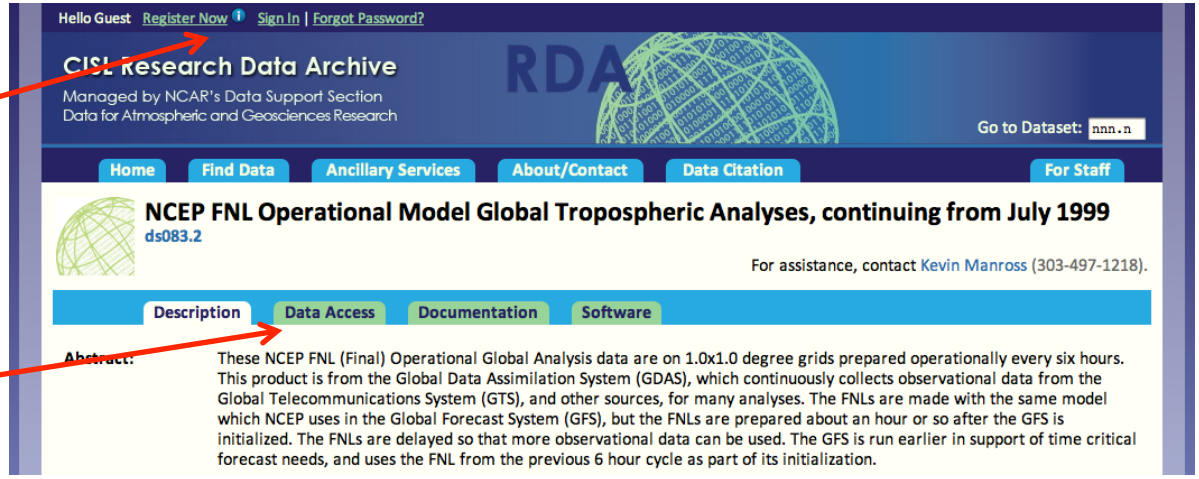
Step 2: Click the dataset you wish to use (for this example, we will use 'FNL from GFS')

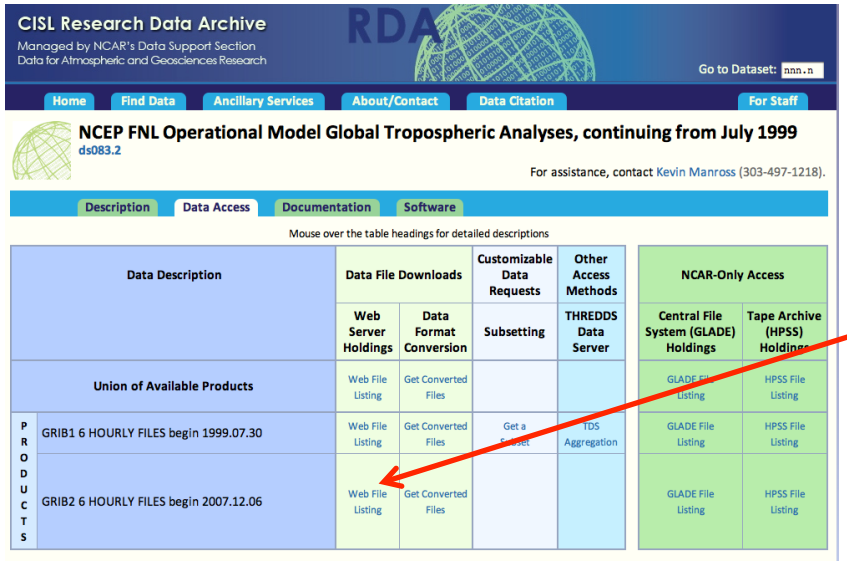
*Note: The NOMADS site has several types of useful data: http://nomads.ncdc.noaa.gov

Download Datasets (continued)

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

Step 4: Click 'Data Access'

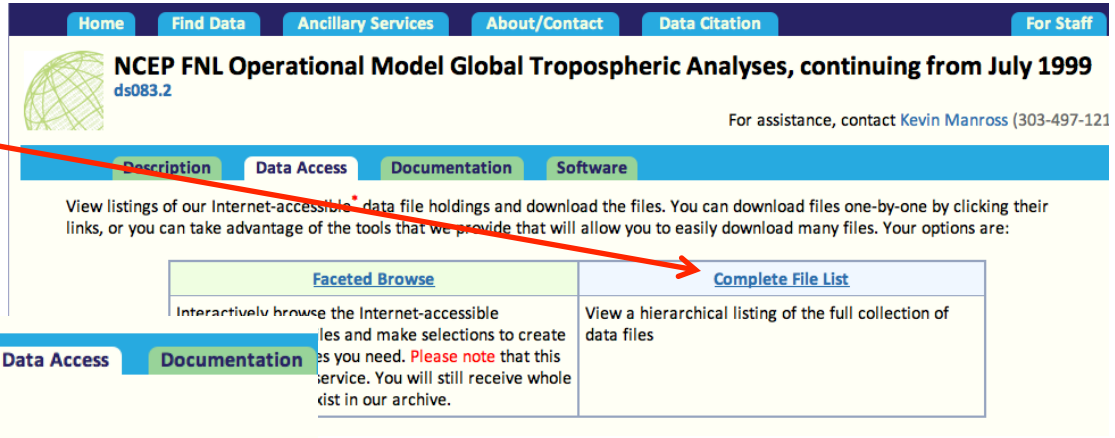


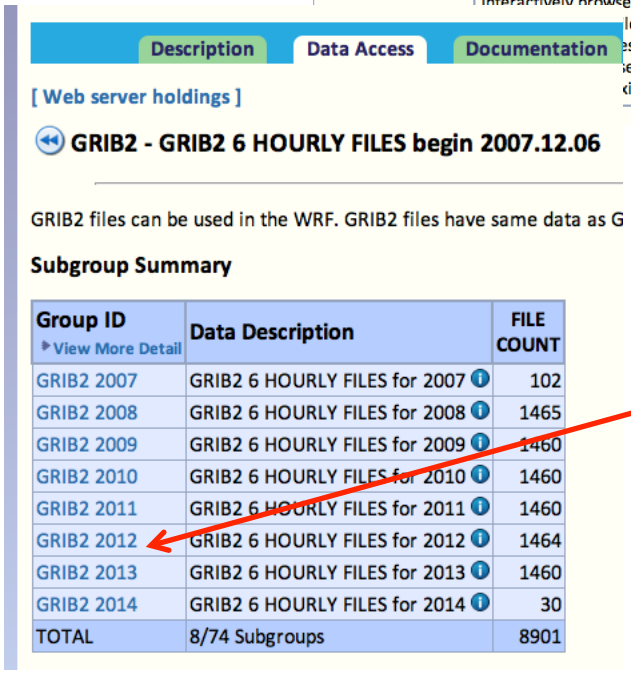


Step 5: Click 'Web File Listing' for the span of years you need

Download Datasets (continued)

Step 6: Click 'Complete File List'

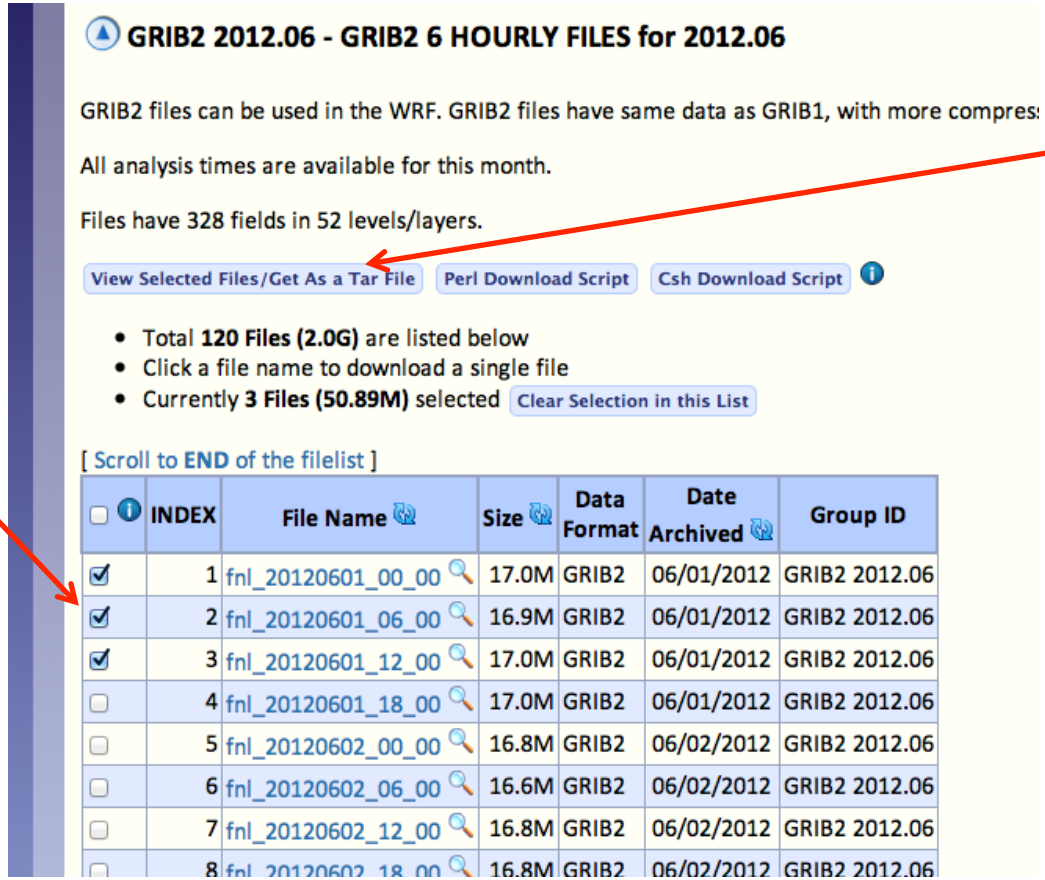




Step 7: Click the year you need. After this, You will click the month you need (not shown)

Download Datasets (continued)

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



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 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

Questions?