Compiling WRF and WPS

Michael Duda

*NCAR is sponsored by the National Science Foundation

Outline

0. Checking system requirements
    - Compilers, UNIX tools
1. Downloading source code and datasets
2. Installing supporting libraries
    - NetCDF, Jasper, zlib, png; optionally, MPI
3. Installing WRF
4. Installing WPS

System requirements

On what kinds of systems will WRF run?

- Generally, any 32- or 64-bit hardware running a UNIX-like operating system
  - Dual-booting into a UNIX-like OS should be fine
- Processor architecture not all that important
  - WRF has been built on POWER, ARM, x86, x86_64 architectures
  - Availability of Fortran and C compilers for the architecture *is* important
- Examples:
  - Laptops, desktops, and clusters running linux
  - Laptops and desktops running MacOS X
  - Clusters running linux or AIX (Cray, IBM, SGI, etc.)
  - *Probably* systems running BSD
  - Your Raspberry Pi

System requirements

On what kinds of systems will WRF *generally not* run?

- Any non-UNIX-like systems
  - Windows, MS-DOS, OS/360, Cygwin, Commodore64
- Systems running linux through a virtual environment
  - Weird filesystem issues have been observed…
- Android and iOS tablets and phones
  - Although this would really impress us if you were to demonstrate otherwise…

System requirements

So much for hardware and OS… what about other software?

- The source code for WRF is written in Fortran and C; therefore, we require Fortran and C compilers
  - We most often test with GNU, PGI, Intel, and XLF compilers
- The configuration and build system for WRF makes use of several shell and scripting languages
  - csh, sh, perl
- Also used throughout the configuration and build system are various UNIX utilities/commands
  - tar, gzip, sed, awk, cut, sort, uname, nm, and many others…

System requirements

Before embarking on the great adventure of compiling WRF, it's important to ensure that your system has *compatible* Fortran and C compilers, shells, and utility programs

The web page

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

serves as a guide through the process of checking system requirements, building libraries, and building WRF and WPS

Outline

0. Checking system requirements

    - Compilers, UNIX tools

1. Downloading source code and datasets

2. Installing supporting libraries

    - NetCDF, Jasper, zlib, png; optionally, MPI

3. Installing WRF

4. Installing WPS

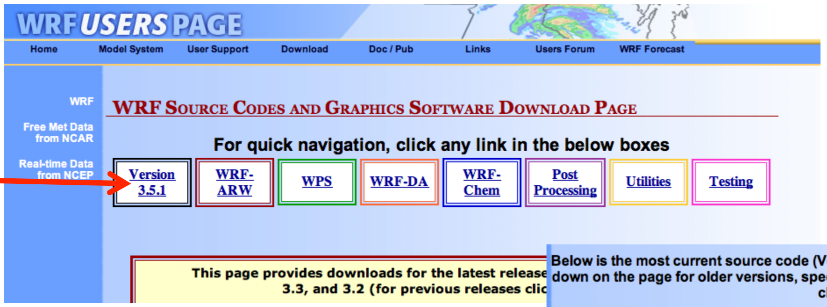
Downloading source code: WRF, WPS

Download WRF and 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 e-mail, and download

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

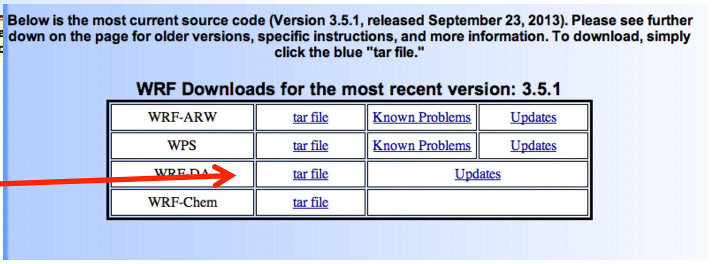


WRF Source Codes and Graphics Software Download Page

For quick navigation, click any link in the below boxes

This page provides downloads for the latest release
3.3, and 3.2 (for previous releases clic

Step 2:
Click on tar files to download



Below is the most current source code (Version 3.5.1, released September 23, 2013). Please see further down on the page for older versions, specific instructions, and more information. To download, simply click the blue "tar file."

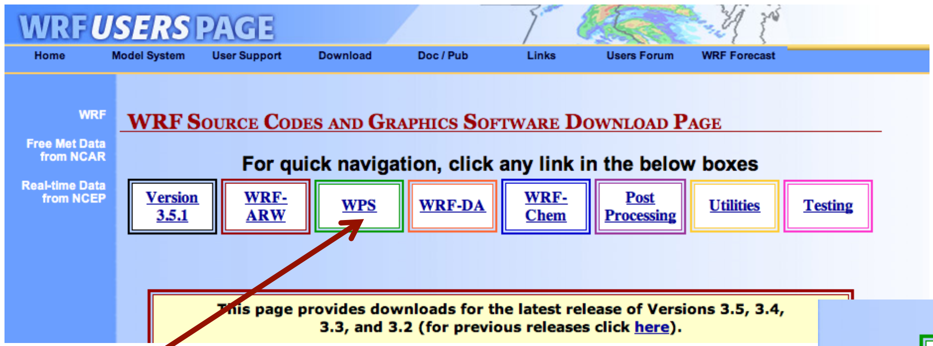
WRF Downloads for the most recent version: 3.5.1

WRF-ARW	tar file	Known Problems	Updates
WPS	tar file	Known Problems	Updates
WRF-DA	tar file	Updates	
WRF-Chem	tar file		

Downloading geographical datasets

From the WRF Download page:

http://www.mmm.ucar.edu/wrf/users/download/get_sources.html



WRF SOURCE CODES AND GRAPHICS SOFTWARE DOWNLOAD PAGE

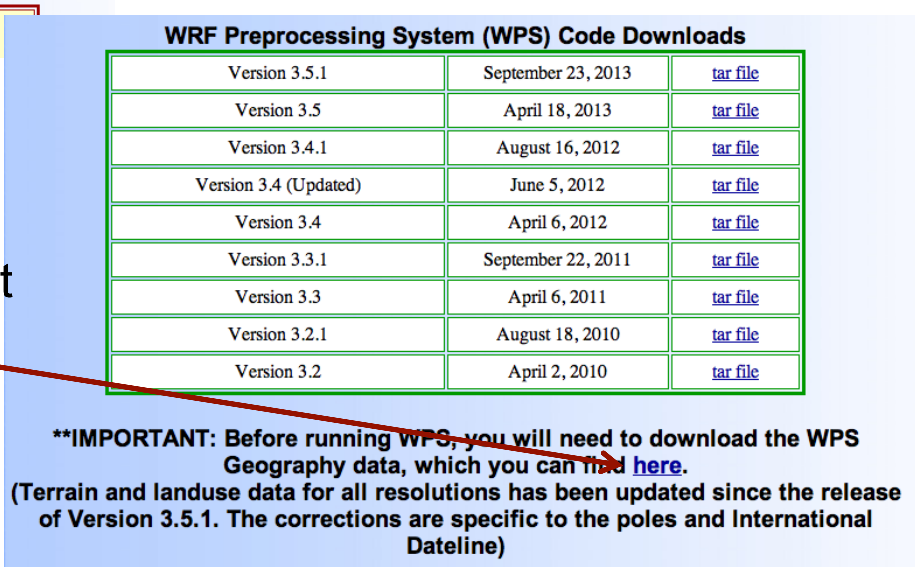
For quick navigation, click any link in the below boxes

Version 3.5.1	WRF-ARW	WPS	WRF-DA	WRF-Chem	Post Processing	Utilities	Testing

This page provides downloads for the latest release of Versions 3.5, 3.4, 3.3, and 3.2 (for previous releases click here).

Step 1: Click 'WPS' box

Step 2: Click 'here' to get geography data



WRF Preprocessing System (WPS) Code Downloads

Version 3.5.1	September 23, 2013	tar file
Version 3.5	April 18, 2013	tar file
Version 3.4.1	August 16, 2012	tar file
Version 3.4 (Updated)	June 5, 2012	tar file
Version 3.4	April 6, 2012	tar file
Version 3.3.1	September 22, 2011	tar file
Version 3.3	April 6, 2011	tar file
Version 3.2.1	August 18, 2010	tar file
Version 3.2	April 2, 2010	tar file

**IMPORTANT: Before running WPS, you will need to download the WPS Geography data, which you can find here.
(Terrain and landuse data for all resolutions has been updated since the release of Version 3.5.1. The corrections are specific to the poles and International Dateline)

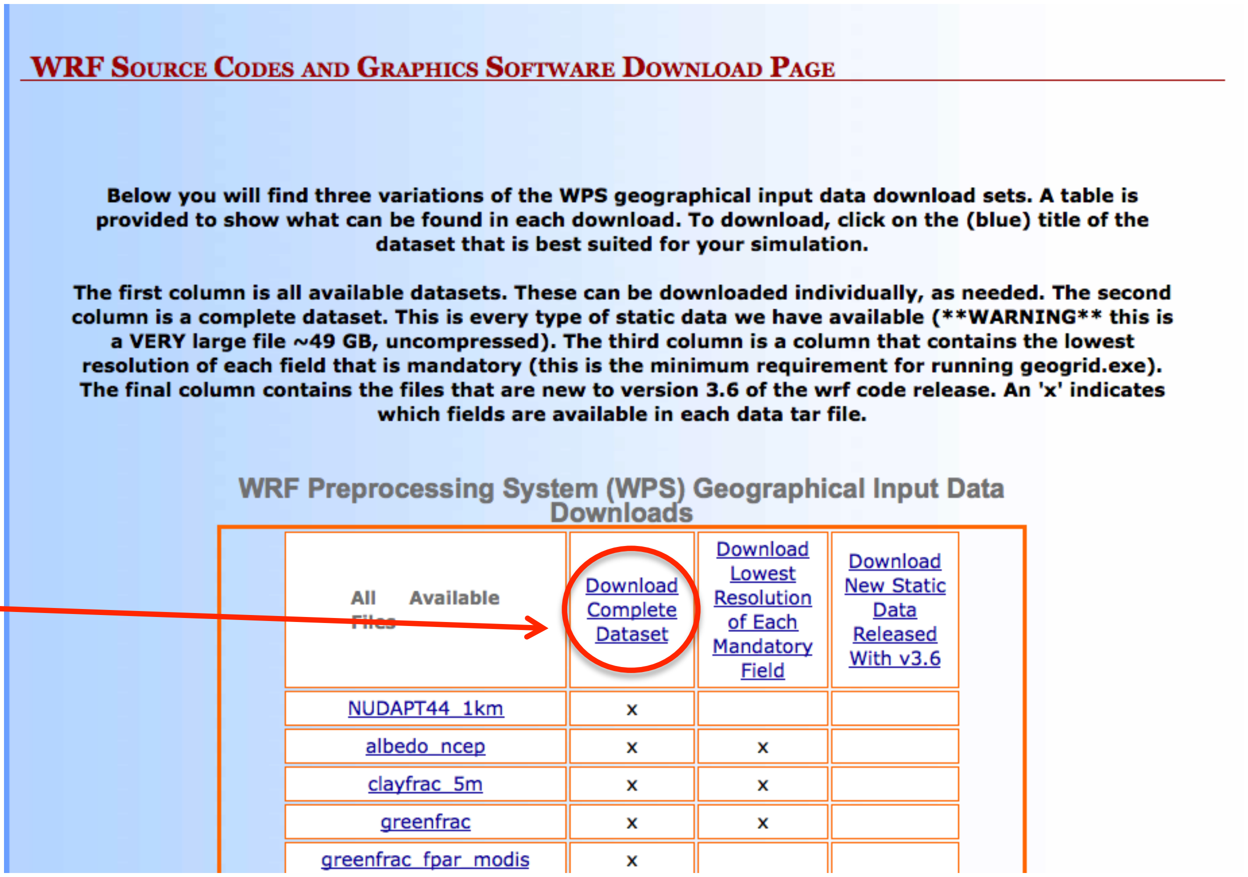
Downloading geographical datasets

Geographical Input and Data Download Page:

http://www.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 Source Codes and Graphics Software Download Page

Below you will find three variations of the WPS geographical input data download sets. A table is provided to show what can be found in each download. To download, click on the (blue) title of the dataset that is best suited for your simulation.

The first column is all available datasets. These can be downloaded individually, as needed. The second column is a complete dataset. This is every type of static data we have available (**WARNING** this is a VERY large file ~49 GB, uncompressed). The third column is a column that contains the lowest resolution of each field that is mandatory (this is the minimum requirement for running geogrid.exe). The final column contains the files that are new to version 3.6 of the wrf code release. An 'x' indicates which fields are available in each data tar file.

WRF Preprocessing System (WPS) Geographical Input Data Downloads

All Available Files	Download Complete Dataset	Download Lowest Resolution of Each Mandatory Field	Download New Static Data Released With v3.6
NUDAPT44_1km	x		
albedo_ncep	x	x	
clayfrac_5m	x	x	
greenfrac	x	x	
greenfrac_fpar_modis	x		

Downloading source code: NetCDF

The only software library that is *required* by WRF is NetCDF

- Can be installed through packages (e.g., apt-get)
- **However, installing from source is easy and ensures compatibility with your Fortran and C compilers!**
  - *We highly recommend installing from source!*

- Download source code from
  http://www.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP2

- Also available from Unidata (http://www.unidata.ucar.edu/, the authors/maintainers of NetCDF)
  - The latest versions are split into different downloads for Fortran and C, and are more difficult to install
  - Using version 4.1.3 from the link above is much easier…

Downloading source code: GRIB2 libraries

In order for the WPS to process GRIB2 files, several compression libraries are required

- The WPS can be compiled without these libraries, but most gridded meteorological data are available on GRIB2

Required libraries: Jasper, PNG, zlib

- Source code for all three libraries available through http://www.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP2

Also available directly from their homepages:
Jasper: http://www.ece.uvic.ca/~frodo/jasper/
PNG: http://libpng.org/pub/png/libpng.html
zlib: http://www.zlib.net/

Downloading source code: MPICH

WRF can be run in parallel using OpenMP, which is supported by most recent Fortran and C compilers

For distributed-memory parallelism, WRF requires an implementation of the MPI-2 standard

- Distributed-memory parallelism required when running across multiple nodes of a non-shared memory cluster
- **We most often test and run WRF using MPICH**
  - Other implementations of MPI exist, e.g., OpenMPI, MVAPICH, vendor-specific implementations

- Download source code from
  http://www.mmm.ucar.edu/wrf/OnLineTutorial/compilation_tutorial.php#STEP2

- MPICH is also available from http://www.mpich.org/

Downloading source code: MPICH

Compute clusters with a queuing system are likely to already have an installation of MPI available; this installation is probably setup to interact with the queuing system to get your job running on available nodes.

Therefore, on such systems, attempting to install MPICH yourself is probably not going to work.

Instead, ensure that the system-supplied MPI tools (mpif90, mpicc, mpiexec, etc.) are in your shell path.

Outline

0. Checking system requirements
    - Compilers, UNIX tools
1. Downloading source code and datasets
2. Installing supporting libraries
    - NetCDF, Jasper, zlib, png; optionally, MPI
3. Installing WRF
4. Installing WPS

Installing supporting libraries: NetCDF

A typical sequence of commands (in csh) to install NetCDF:

```
setenv NETCDF path_to_where_netcdf_will_be_installed
setenv CC gcc
setenv CXX g++
setenv FC gfortran
setenv FCFLAGS -m64     # FCFLAGS may be needed
setenv F77 gfortran
setenv FFLAGS -m64      # FFLAGS may be needed

tar xzvf netcdf-4.1.3.tar.gz
cd netcdf-4.1.3
./configure --prefix=$NETCDF --disable-dap  \
          --disable-netcdf-4 --disable-shared
make
make install
setenv PATH $NETCDF/bin:$PATH    # adds ncdump command to path
```

Installing supporting libraries: zlib

A typical sequence of commands (in csh) to install zlib:

```
setenv GRIB2 path_to_where_grib2_libs_will_be_installed
setenv CC gcc

tar xzvf zlib-1.2.7.tar.gz
cd zlib-1.2.7
./configure --prefix=$GRIB2
make
make install
```

Note: Installing Jasper, PNG, and zlib in a common directory (i.e., `$GRIB2`) will simplify the configuration of the WPS

Installing supporting libraries: png

A typical sequence of commands (in csh) to install png:

```
setenv GRIB2 path_to_where_grib2_libs_will_be_installed
setenv CC gcc
setenv CPPFLAGS "-I${GRIB2}/include"  # for locating zlib
setenv LDFLAGS "-L${GRIB2}/lib"       # for locating zlib

tar xzvf libpng-1.2.50.tar.gz
cd libpng-1.2.50
./configure --prefix=$GRIB2
make
make install
```

Installing supporting libraries: Jasper

A typical sequence of commands (in csh) to install Jasper:

```
setenv GRIB2 path_to_where_grib2_libs_will_be_installed
setenv CC gcc

tar xzvf jasper-1.900.1.tar.gz
cd jasper-1.900.1
./configure --prefix=$GRIB2
make
make install
```

With all libraries for GRIB2 compression installed in `$GRIB2`, setting

```
setenv JASPERINC ${GRIB2}/include
setenv JASPERLIB ${GRIB2}/lib
```

will allow the WPS configure script to find these libraries.

Installing supporting libraries: MPICH

A typical sequence of commands (in csh) to install MPICH:

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

```
setenv MPICH path_to_where_mpich_will_be_installed
setenv CC gcc
setenv CXX g++
setenv FC gfortran
setenv FCFLAGS -m64     # FCFLAGS may be needed
setenv F77 gfortran
setenv FFLAGS -m64      # FFLAGS may be needed

tar xzvf mpich-3.0.4.tar.gz
cd mpich-3.0.4
./configure --prefix=$MPICH
make
make install
setenv PATH $MPICH/bin:$PATH  # add mpif90, mpicc to path
```

Outline

0. Checking system requirements
   - Compilers, UNIX tools
1. Downloading source code and datasets
2. Installing supporting libraries
   - NetCDF, Jasper, zlib, png; optionally, MPI
3. Installing WRF
4. Installing WPS

Installing WRF

Before you *even think about* configuring and compiling WRF, ensure that the environment variable `NETCDF` is set to the root installation of the NetCDF library

- Running `ls $NETCDF` should show directories `bin`, `include`, `lib`, `share`

If you plan to compile WRF with support for distributed-memory parallelism, ensure that the MPI *compiler wrappers* are in your path

- Check with `which mpif90` and `which mpicc`

Unpack the WRFV3.TAR.gz file and change to the resulting 'WRFV3' directory:

```
tar xzvf WRFV3.TAR.gz
cd WRFV3
```

Installing WRF

It is important to compile WRFV3 **first**, before WPS
WPS makes use of the external I/O libraries in the *WRFV3/external* directory that are built when WRF is installed

Compiling WRF is a two-step process:

1) Create a configuration file for your computer and compiler

```
./configure
```

2) Compile the code

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

Installing WRF

From inside the WRFV3 directory, run `./configure`

```
Will use NETCDF in dir: /glade/u/home/duda/libs_intel/netcdf
Will use PNETCDF in dir: /glade/u/home/duda/libs_intel/parallel-netcdf
PHDF5 not set in environment. Will configure WRF for use without.

------------------------------------------------------------------------
Please select from among the following supported platforms.

 1.  Linux x86_64 i486 i586 i686, PGI compiler with gcc  (serial)
 2.  Linux x86_64 i486 i586 i686, PGI compiler with gcc  (smpar)
 3.  Linux x86_64 i486 i586 i686, PGI compiler with gcc  (dmpar)
 4.  Linux x86_64 i486 i586 i686, PGI compiler with gcc  (dm+sm)
…
32.  x86_64 Linux, gfortran compiler with gcc   (serial)
33.  x86_64 Linux, gfortran compiler with gcc   (smpar)
34.  x86_64 Linux, gfortran compiler with gcc   (dmpar)
35.  x86_64 Linux, gfortran compiler with gcc   (dm+sm)
…
Enter selection [1-47] : 34
------------------------------------------------------------------------
Compile for nesting? (1=basic, 2=preset moves, 3=vortex following) [default 1]: 1

Configuration successful. To build the model type compile .
```

Installing WRF

The configuration step should yield a `configure.wrf` file

- Modifications to compiler flags (optimization, debugging, etc.) can be made by editing this file

To compile WRF, run

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

where *`test_case`* is one of the following:

$$\left.\begin{array}{l}\text{em\_quarter\_ss}\\ \text{em\_b\_wave}\\ \text{em\_les}\\ \text{em\_heldsuarez}\\ \text{em\_tropical\_cyclone}\end{array}\right\}\text{2d ideal}$$

em_real (3d real)

$$\left.\begin{array}{l}\text{em\_hill2d\_x}\\ \text{em\_squall2d\_x}\\ \text{em\_squall2d\_y}\\ \text{em\_grav2d\_x}\\ \text{em\_seabreeze2d\_x}\end{array}\right\}\text{2d ideal}$$

em_scm_xy (1d ideal)

Start to make a pot of coffee… compilation may take 20 – 30 minutes!

Installing WRF

If compilation is successful, the following executables should be found in the `WRFV3/main` directory

- Real-data case:

`wrf.exe`	-- model executable
`real.exe`	-- real-data initialization
`ndown.exe`	-- off-line, one-way nesting
`tc.exe`	-- TC bogussing, serial only

- Idealized case:

`wrf.exe`	-- model executable
`ideal.exe`*	-- idealized initialization

*All idealized cases produce an executable named ideal.exe, which must be recompiled when changing between idealized cases

Executables will by symbolically linked to the `WRFV3/run` and `WRFV3/test/`*`test_case`* (e.g., `WRFV3/test/em_real`) directories; WRF may be run from either location.

Installing WRF

If compilation was unsuccessful, try searching for `Error` in the `log.compile` file

If you do happen to find this error:

```
/* Copyright (C) 1991-2013 Free Software Foundation, Inc.
   1
   Error: Non-numeric character in statement label at (1)
   fail.f:1.2:
```

please let us know (e.g, in person this week, or via wrfhelp@ucar.edu); the error is caused by newer versions of CPP that insert C header files into pre-processed files.

Outline

0. Checking system requirements

    - Compilers, UNIX tools

1. Downloading source code and datasets

2. Installing supporting libraries

    - NetCDF, Jasper, zlib, png; optionally, MPI

3. Installing WRF

4. Installing WPS

Installing WPS

Before configuring WPS, ensure that the JASPERLIB and JASPERINC environment variables are set to the root installation directory of GRIB2 libraries

- `ls $JASPERLIB/` should show `libjasper.a`, `libpng.a`, and `libz.a`
- `ls $JASPERINC/` should show `jasper/`, `png.h`, and `zlib.h`

Unpack the WPSV3.TAR.gz file and change to the resulting 'WPS' directory:

```
tar xzvf WPSV3.TAR.gz
cd WPS
```

The WPS configuration assumes that the WRFV3 directory and the WPS directory are both contained inside the same parent directory; alternate paths to the WRFV3 directory can be manually set after the configuration step

Installing WPS

From inside the WPS directory, run `./configure`

```
Will use NETCDF in dir: /glade/u/home/duda/libs_intel/netcdf
Found Jasper environment variables for GRIB2 support...
  $JASPERLIB = /glade/u/home/wrfhelp/UNGRIB_LIBRARIES/lib
  $JASPERINC = /glade/u/home/wrfhelp/UNGRIB_LIBRARIES/include
------------------------------------------------------------------------
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)
  …
  33.  Cray XC CLE/Linux x86_64, Intel compiler   (serial)
  34.  Cray XC CLE/Linux x86_64, Intel compiler   (serial_NO_GRIB2)
  35.  Cray XC CLE/Linux x86_64, Intel compiler   (dmpar)
  36.  Cray XC CLE/Linux x86_64, Intel compiler   (dmpar_NO_GRIB2)

Enter selection [1-36] : 1
```

Installing WPS

The configuration step should yield a `configure.wps` file

- You can edit this file to:
  - Set the path where WRF is installed, if not `../WRFV3`
  - Use other compiler flags (optimization, debugging, etc.)
  - Demonstrate your WRF prowess to colleagues

To compile the WPS, run

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

If compilation is successful, the following executables should be found in the top-level WPS directory

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

Cleaning a WRF/WPS installation

Both the WPS and WRF provide a script, `clean`, which is the opposite of `compile`

- Removes object files, module files, auto-generated code, and (optionally) executable files

Under what circumstances would we need to run `./clean`?

- **Generally, we run “./clean -a”. The “-a” option means to clean all files.**
- Run `./clean -a` after editing the Registry file
- Run `./clean -a` before recompiling WRF with different compiler options

However, there is no need to run `./clean` when editing regular source files (i.e., `*.F` and `*.c` files)

Questions?