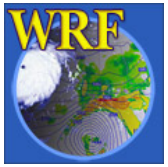
Installing WPS & WRF

Wei Wang

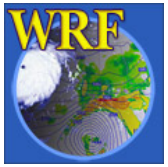
NCAR/NESL/MMM

November 2011



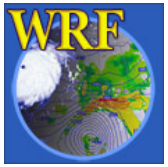
Installing Steps

- Check system requirements
- Download source code tar files
- Download datasets
- Compile WRFV3 first
- Compile WPS



Check System Requirements

- Required libraries
  - **NetCDF** (needed by WRF and WPS)
  - **NCAR Graphics** (*optional but can be helpful* – used by graphical utility programs)
- Optional libraries for GRIB2 met data support
  - **JasPer** (JPEG 2000 "lossy" compression library)
  - **PNG** ("lossless" compression library)
  - **zlib** (compression library used by PNG)
- Optional MPI library:
  - Common ones: mpich, mpich2
  - Less common: openmpi



Check System Requirements

- Installation of these libraries is *not* part of the WPS and WRF installation scripts
  - We recommend having a system administrator install the required libraries before installing WRF or WPS
- Make sure that these libraries are installed using the compilers available to you to compile WRF/WPS code.

Download WPS & WRF Source Code tar Files

- Download WPS & WRF source code from

  http://www.mmm.ucar.edu/wrf/users/downloads.html

  Click 'WRF' on the side menu, then

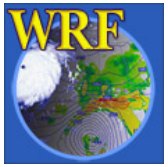
  > 'New Users', register and download, or

  > 'Returning Users', your email and download

- Get the latest released codes:

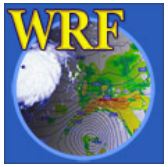
  `WPSV3.TAR.gz`

  `WRFV3.TAR.gz`



Additional Downloads

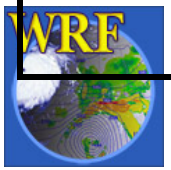
- Test datasets
  - WPS output for WRF; can be useful for testing
  - Sample grib data for WPS
- Terrain, land state datasets for geogrid
  - Full resolution (30”, 2’, 5’, 10’ version)
  - Lower resolution (10 minutes version)
- Download from the same site as the source code tar files.



Static Terrestrial Data

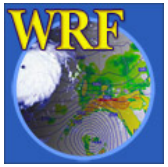
The `geog.tar.gz` file contains the following data (~ 13 GB when uncompressed):

albedo_ncep	monthly surface albedo
greenfrac	monthly vegetation fraction
maxsnowalb	maximum snow albedo
landuse USGS	24+1 categories, 30”, 2’, 5’ and 10’ (since V3.3, it includes ‘lake’ category)
landuse MODIS	20+1 categories, 30”, new in V3.1, Noah LSM only, ‘lake’ added in V3.3



Static Terrestrial Data

soiltemp	annual mean deep soil temperature
soiltype_top	top-layer soil type, 30”, 2’, 5’, 10’
soiltype_bot	bot-layer soil type, 30”, 2’, 5’, 10’
topo	topography, 30”, 2’, 5’, 10’
orogwd	subgrid orography informationfor gravity wave drag option, new in V3.1
islope	slope index (not used)



Static Terrestrial Data

- Low resolution set available (398 MB only; 10', ~18 km, resolutions).
- Since the full resolution dataset is big, it should be placed in a common location so that multiple users can share

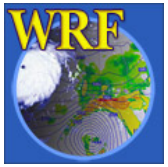
Unzip and Untar tar Files

- Create a working directory, and uncompress both WPS and WRF tar files:

```
gunzip WPSV3.TAR.gz
tar -xf WPSV3.TAR
gunzip WRFV3.TAR.gz
tar -xf WRFV3.TAR
```

- After unzip and untar, you should have these directories in your working directory:

```
WPS/
WRFV3/
```



WPS/ Directory

README

clean
compile
configure
} compile, clean scripts

arch/ — compile rules

geogrid/
ungrib/
metgrid/
} source code directories

util/ — utilities

link_grib.csh

namelist.wps
namelist.wps_all-options
} runtime options

WRFV3/ Directory

```
Makefile
README
README_test_cases
clean
compile
configure
Registry/
arch/
dyn_em/
dyn_exp/
external/
frame/
inc/
main/
phys/
share/
tools/
run/
test/
```

compile scripts

data dictionary

compile rules

source code directories

run directories

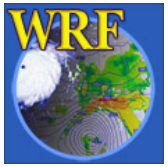
Before compiling..

- Check where your netCDF *library* and *include* directories are
- If it is not in the usual location, i.e.

  /usr/local/netcdf

  Use NETCDF environment variable to set the path. For C-shell environment,

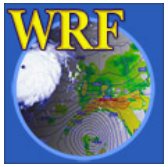
  `setenv NETCDF /where-netcdf-is`



Before Compiling..

- Know how your netCDF library is installed.
  - what compiler is used
- As a general rule, the netCDF library needs to be installed using the same compiler as the one used to compile WRF and WPS

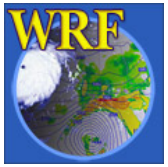
  e.g. PGI compiler



Compiling WRFV3 first

Why?

- WPS makes use of the external I/O libraries in WRFV3/external directory
- These libraries are built when WRF is installed



How to Compile WRFV3?

There are two steps:

1) Create a configuration file for your computer and compiler

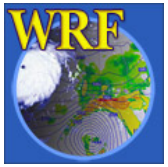
```
./configure
```

2) Compile the code

```
./compile test_case
```

To start over, type

```
clean -a
```



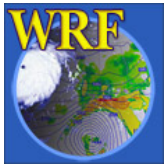
Creating a configuration file

Step 1: type

`./configure`

This is a script that checks the system hardware and software (mostly *netCDF*), and then offers a user a number of compile choices:

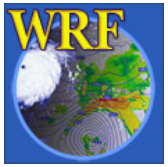
- Serial, OpenMP (smpar), MPI (dmpar), MPI +OpenMP (dm+sm)
- Type of nesting (no nesting, basic, preset moves, vortex following)



Notes on Compile

- If MPI or OpenMP is used, make sure that you have the parallel libraries on the computer
  - mpich
  - smp (typically comes with the Linux)
- If anticipating writing large netCDF files, do the following before compiling:

```
setenv WRFIO_NCD_LARGE_FILE_SUPPORT 1
```

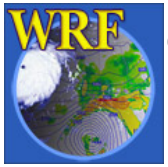


Running configuration script: *type of compile*

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /usr/local/netcdf
PHDF5 not set in environment. Will configure WRF for use without.
configure:  WRF operating system set to "Linux" via environment variable $WRF_OS
configure:  WRF machine set to "i686" via environment variable $WRF_MACH
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
------------------------------------------------------------------------
Please select from among the following supported platforms.

   1.  Linux i486 i586 i686, gfortran compiler with gcc  (serial)
   2.  Linux i486 i586 i686, gfortran compiler with gcc  (smpar)
   3.  Linux i486 i586 i686, gfortran compiler with gcc  (dmpar)
   4.  Linux i486 i586 i686, gfortran compiler with gcc  (dm+sm)
   5.  Linux i486 i586 i686, g95 compiler with gcc  (serial)
   6.  Linux i486 i586 i686, g95 compiler with gcc  (dmpar)
   7.  Linux i486 i586 i686, PGI compiler with gcc  (serial)
   8.  Linux i486 i586 i686, PGI compiler with gcc  (smpar)
   9.  Linux i486 i586 i686, PGI compiler with gcc  (dmpar)
  10.  Linux i486 i586 i686, PGI compiler with gcc  (dm+sm)
  11.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (serial)
  12.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (smpar)
  13.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (dmpar)
  14.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (dm+sm)

Enter selection [1-16] :
```

Running configuration script: *nesting options*

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /usr/local/netcdf
PHDF5 not set in environment. Will configure WRF for use without.
configure:  WRF operating system set to "Linux" via environment variable $WRF_OS
configure:  WRF machine set to "i686" via environment variable $WRF_MACH
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
------------------------------------------------------------------------
Please select from among the following supported platforms.

   1.  Linux i486 i586 i686, gfortran compiler with gcc  (serial)
   2.  Linux i486 i586 i686, gfortran compiler with gcc  (smpar)
   3.  Linux i486 i586 i686, gfortran compiler with gcc  (dmpar)
   4.  Linux i486 i586 i686, gfortran compiler with gcc  (dm+sm)
   5.  Linux i486 i586 i686, g95 compiler with gcc  (serial)
   6.  Linux i486 i586 i686, g95 compiler with gcc  (dmpar)
   7.  Linux i486 i586 i686, PGI compiler with gcc  (serial)
   8.  Linux i486 i586 i686, PGI compiler with gcc  (smpar)
   9.  Linux i486 i586 i686, PGI compiler with gcc  (dmpar)
  10.  Linux i486 i586 i686, PGI compiler with gcc  (dm+sm)
  11.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (serial)
  12.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (smpar)
  13.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (dmpar)
  14.  Linux x86_64 i486 i586 i686, ifort compiler with icc  (dm+sm)

Enter selection [1-16] : 9
------------------------------------------------------------------------
Compile for nesting? (1=basic, 2=preset moves, 3=vortex following) [default 1]:
```

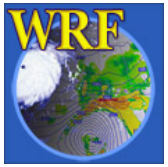


Create a configuration file

The result of running the **`configure`** script is the generation of a file called:

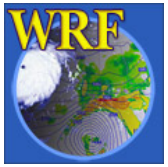
`configure.wrf`

This file contains compilation options, rules etc. specific to your computer.



Sample of what is inside a `configure.wrf` file

```
FC              =       pgf90
LD              =       pgf90
CC              =       gcc -DFSEEKO64_OK
SCC             =       $(CC)
RWORDSIZE       =       $(NATIVE_RWORDSIZE)
SFC             =       $(FC)
CFLAGS          =
FCOPTIM         =       -O2 # -fast
FCDEBUG         =       #-g
FCBASEOPTS      =       -w -byteswapio -Mfree
                        -tp p6 $(FCDEBUG)
FCFLAGS         =       $(FCOPTIM) $(FCBASEOPTS)
```



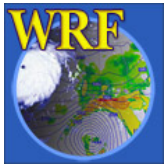
What consists of a `configure.wrf` file

The **`configure.wrf`** file is built from three pieces from **`arch/`** directory:

1) **`preamble_new`**: uniform requirement for the code, such as maximum number of domains, word size, etc.
2) **`configure_new.defaults`**: selection of compiler, parallel, runtime system library (RSLs)
3) **`postamble_new`**: standard make rules and dependencies.

   `preamble/postamble` do not change

4) **`noopt_exceptions`**: list of routines compiled with low optimization options



How to Compile?

Step 2: type

`./compile test_case` or

`./compile test_case >& compile.log`

where ***`test_case`*** is one of the following:
(type **`./compile`** to find out)

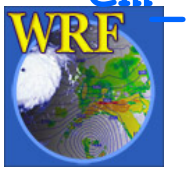
`em_real` 3d real

3d ideal:
- `em_quarter_ss`
- `em_b_wave`
- `em_les`
- `em_heldsuarez`

`em_tropical_cyclone` (new in V3.3)

2d ideal:
- `em_hill2d_x`
- `em_squall2d_x`
- `em_squall2d_y`
- `em_grav2d_x`
- `em_seabreeze2d_x`

1d ideal:
- `em_scm_xy`

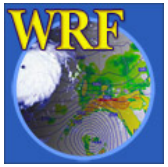


More on Compile

- Compiling WRF code will take 20 - 30 min, depending on options chosen
- Since V3.2, parallel compile is supported if 'make' on the computer supports it
- Two processors are used in default compile. If you would like to change it, set the following environment variable before compile

```
setenv J "-j 1"
```

-- change to use only one processor

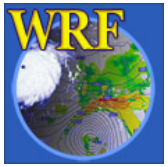


Make change for your system

- If netCDF is not in **`/usr/local`**, you can use environment variable NETCDF to set the path to netCDF before typing **`'configure'`**. e.g. on a Linux with PGI-compiled netCDF:

  `setenv NETCDF /usr/local/netcdf-pgi`

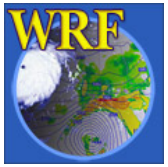
- If you use a Linux, a number of compiler may be available (PGI, Intel, g95). As a general rule, make sure your netCDF and MPI libraries are installed using the same compiler you use to compile WRF.



Make change for your system

- One may edit **`configure.wrf`** to make changes for your system
- If option for your system is not available, add one to **`arch/configure_new.defaults`**

  Start with something close to your system from the file, and serial compile



WRF executables: names and locations

If compile is successful, you should find these executables in **`WRFV3/main/`**.

If you compile for a real data case:

- **`wrf.exe`** - model executable
- **`real.exe`** - real data initialization
- **`ndown.exe`** - one-way nesting
- **`tc.exe`** - for tc bogusing (serial only)

If you compile an ideal case, you should have:

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

→ each ideal test case compile creates a different **`ideal.exe`**

WRF executables: names and locations

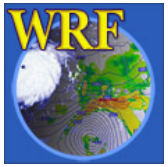
These executables are linked to:

`WRFV3/run`

and

`WRFV3/test/em_`*`test_case`*

➔ One can go to either directory to run.



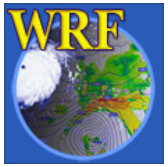
WRFV3/run directory

```
LANDUSE.TBL
ETAMPNEW_DATA
RRTM_DATA
SOILPARM.TBL
VEGPARM.TBL
urban_param.tbl
tr49t67
tr49t85
tr67t85
gribmap.txt
grib2map.tbl
```

these files are for model physics use, and reside in this directory

```
namelist.input -> ../test/test_case/namelist.input
real.exe -> ../main/real.exe
wrf.exe -> ../main/wrf.exe
ndown.exe -> ../main/ndown.exe
```

…. (a few more)



WRFV3/test/em_*real* directory

```
LANDUSE.TBL -> ../../run/LANDUSE.TBL
ETAMPNEW_DATA -> ../../run/ETAMPNEW_DATA
RRTM_DATA -> ../../run/RRTM_DATA
SOILPARM.TBL -> ../../run/SOILPARM.TBL
VEGPARM.TBL -> ../../run/VEGPARM.TBL
urban_param.tbl -> ../../run/urban_param.tbl
tr49t67 -> ../../run/tr49t67
tr49t85 -> ../../run/tr49t85
tr67t85 -> ../../run/tr67t85
gribmap.txt -> ../../run/gribmap.txt
grib2map.tbl -> ../../run/grib2map.tbl
namelist.input          - require editing
real.exe -> ../../main/real.exe
wrf.exe -> ../../main/wrf.exe
ndown.exe -> ../../main/ndown.exe
```

…. (a few more)

How to Compile WPS?

Once WRFV3 is compiled, change directory to WPS to compile WPS.

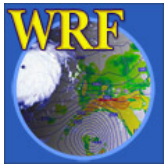
There are two steps here too:

1) Create a configuration file for your computer

```
./configure
```

2) Compile the code

```
./compile
```



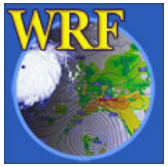
Creating a configuration file

Step 1: type

`./configure`

This is a script that checks the system hardware and software (mostly *netCDF*), and then offers a user a number of compile choices:

- Serial, or MPI (serial usually sufficient)
- Whether to compiling GRIB 2 (requires additional external libraries: *zlib, jasper* and *png*)



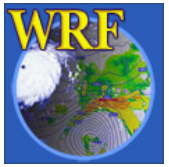
Creating a configuration file

The result of running the **`configure`** script is the generation of a file called:

`configure.wps`

This file contains compilation options, rules etc. specific to your computer.

One may compile WRF model with MPI, but one can compile WPS using serial option, unless one is using very large domains.



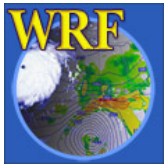
How to Compile?

Step 2: type

`./compile` or

`./compile >& compile.log &` (recommended)

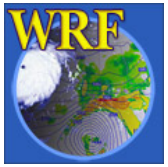
(it doesn't take very long to compile WPS)



WPS executables

If compile is successful, you should find these executables created in WPS/ directory (and they are linked, respectively, to the their source code directories),

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

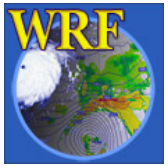


WPS utility executables

If compile is successful, you should also find these executables in WPS/util directory,

`util/plotgrids.exe` - plot a domain map
`util/g1print.exe` - print grib 1 data
`util/g2print.exe` - print grib 2 data
`util/rd_intermediate.exe`
- print data information from ungrib output
`util/plotfmt.exe` - plot intermediate file

Note `plotgrids.exe` and `plotfmt.exe` build require NCAR Graphics



WPS utility executables

More utilities in WPS/util directory,

`util/avg_tsfc.exe`

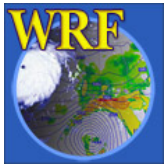
- compute average surface temp to use as substrate temp for 5-layer soil model option, or skin temp if it is not available, or provide lake temp

`util/mod_levs.exe`

- remove pressure levels from intermediate files

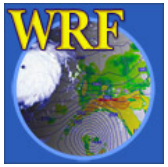
`util/calc_ecmwf_p.exe`

- calculate height, pressure and RH for ECWMF model-level data



Common Problems with Installation

- Executables do not exist
  - Check the location of netCDF library
  - See if netCDF is installed with the same compiler that you use to compile WRF/WPS
  - Try simple compile option first. e.g. serial
  - Repeat the compile procedure starting by typing '**`clean -a`**'



Resources

- Information on compiling WRF and WPS can be found in the WRF User's Guide, Chapter 2, 3 and 5