Installing and Running WPS & WRF

For WRF Version 3.2 and 3.3

Wei Wang

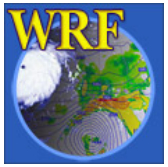
NCAR/NESL/MMM

Installing Steps

- Check system requirements
- Download source codes
- Download datasets
- Compile WRFV3 first
- Compile WPS

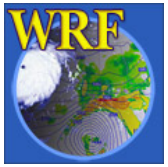
Check System Requirements

- Required libraries
  - **NetCDF** (needed by WRF and WPS)
  - **NCAR Graphics** (*optional but recommended* – 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 Codes

- Download WPS & WRF source codes from

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

  Click 'WRF' on the side menu, then

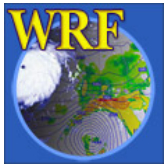
  > 'New Users', register and download, or

  > 'Returning Users', your email and download

- Get the latest released codes:

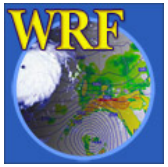
  `WPSV3.TAR.gz`

  `WRFV3.TAR.gz`



Additional Downloads

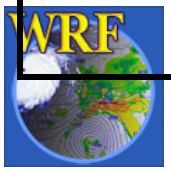
- Test datasets
  - WPS output for WRF; can be useful for testing
  - Sample grib data for WPS
- Terrain, land state datasets for geogrid
  - Full resolution (30”, 2’, 5’, 10’ version)
  - Lower resolution (10 minutes version)
- Download from the same site as the source codes.



Static Terrestrial Data

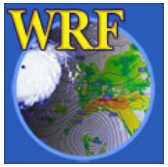
The `geog.tar.gz` file contains the following data (~ 13 GB when uncompressed):

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



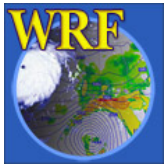
Static Terrestrial Data

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 common location so that multiple users can share



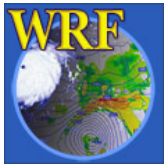
Unzip and Untar tar Files

- 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 opt

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

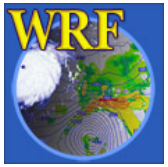
Before compiling..

- Check where your netCDF library and include file 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,

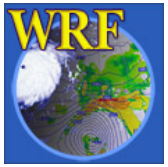
  `setenv NETCDF /where-netcdf-is`



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 one uses to compile WRF and WPS codes

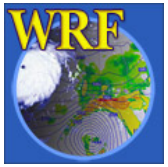
  e.g. PGI compiler



Compile 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?

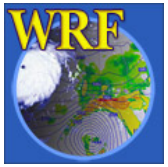
There are two steps:

1) Create a configuration file for your computer and compiler

   `./configure`

2) Compile the code

   `./compile` ***`test_case`***



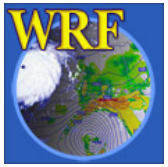
Create configuration file

Step 1: type

`./configure`

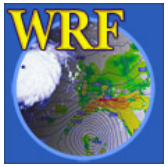
This is a script that checks the system hardware and software (mostly *netCDF*), and then offers a user a number of compile choices:

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



If using any parallel compiling option

- If MPI or OpenMP is used, make sure that you have the parallel libraries on the computer
  - mpich
  - smp

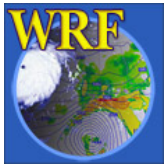


Running configuration script: *type of compile*

```
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] :
```

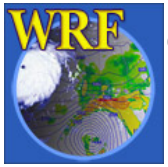


Running configuration script: *nesting options*

```
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]:
```

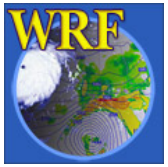


Create a configuration file

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)
```

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)

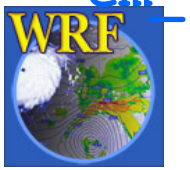
`em_real` 3d real

`em_quarter_ss`
`em_b_wave`
`em_les`
`em_heldsuarez`
} 3d ideal

`em_tropical_cyclone`
`(new in V3.3)`

`em_hill2d_x`
`em_squall2d_x`
`em_squall2d_y`
`em_grav2d_x`
`em_seabreeze2d_x`
} 2d ideal

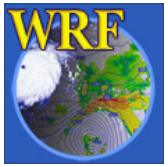
`em_scm_xy` 1d ideal



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



WRF executables: names and locations

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

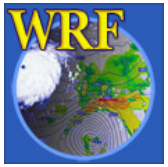
If you compile for a real data case:

- **`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 executable



WRF executables: names and locations

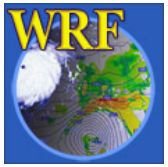
These executables are linked to:

`WRFV3/run`

and

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

-> One can go to either directory to run.



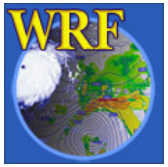
WRFV3/run directory

```
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

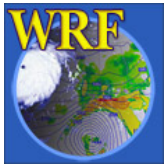
There are two steps here too:

1) Create a configuration file for your computer

```
./configure
```

2) Compile the code

```
./compile
```



Create 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*)

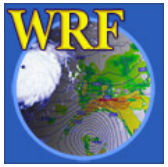
Create a configuration file

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 compile WPS using serial option unless one is using very large domains.



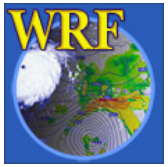
How to Compile?

Step 2: type

`./compile` or

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

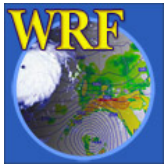
(it doesn't take very long to compile WPS)



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
```

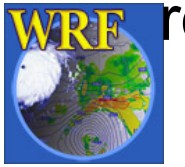


WPS utility executables

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`

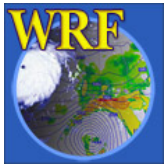
- compute average surface temp to use as substrate temp for 5-layer soil model option or skin temp if it is not available

`util/mod_levs.exe`

- remove pressure levels from intermediate files

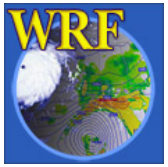
`util/calc_ecmwf_p.exe`

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



Common Problems with Installation

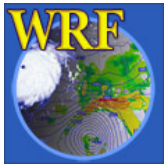
- Executables do not exist
  - Check the location of netCDF library
  - See if netCDF is installed with the same compiler that you use to compile WRF/WPS
  - Try simple compile option first



Running a Real Data Case

Steps to Run WPS

1. Go to *WPS/*
2. Edit *namelist.wps* for your case
3. Run *geogrid.exe* to set up domain
   -- Run *plotgrids.exe* to configure your domain
   (or use *plotgrids.ncl*)
4. Run *ungrid.exe* to degrib met data
5. Run *metgrid.exe* to interpolate met data to model grid



A note on namelist

- A Fortran namelist contains a list of *runtime* options for the program to read in during its execution. Use of a namelist allows one to change runtime configuration without the need to recompile the source code.
- Fortran 90 namelist has very specific format, so edit with care:

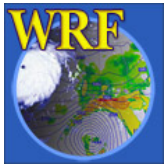
  `&namelist-record` - start

  `/` - end

- As a general rule:

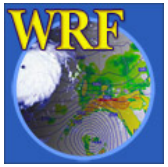
  Multiple columns: domain dependent

  Single column: value valid for all domains



example of a partial WPS namelist

```
&share
 wrf_core = 'ARW',
 max_dom = 2,
 start_date = '2006-08-16_12:00:00','2006-08-16_12:00:00',
 end_date   = '2006-08-16_18:00:00','2006-08-16_12:00:00',
 interval_seconds = 21600
 io_form_geogrid = 2,
/
&geogrid
 parent_id         =   1,   1,
 parent_grid_ratio =   1,   3,
 e_we              =  74, 112,
 e_sn              =  61,  97,
 geog_data_res     = '10m','2m',
 dx = 30000,
 dy = 30000,
 map_proj = 'lambert',
 geog_data_path = '/mmm/users/wrfhelp/WPS_GEOG'
/
&ungrib
 out_format = 'WPS',
 prefix = 'FILE',
/
&metgrid
 fg_name = 'FILE'
 io_form_metgrid = 2,
/
```

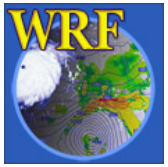


Running geogrid

- Edit namelist records **`&share`** and **`&geogrid`**
- make sure GEOGRID.TBL is linked to GEOGRID.TBL.ARW (by default, it is)
- Type the following to run:

  `./geogrid.exe`
- If successful, you should see

```
Successful completion of geogrid
```



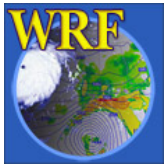
Running geogrid

- Output from geogrid:

  geo_em.d01.nc

  geo_em.d02.nc (for a nest)

- Use tools like **ncview** to quickly check the output



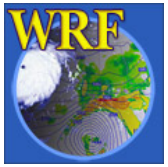
Running ungrib

- Edit namelist record **&share** (for dates) and **&ungrib** in **namelist.wps**
- Link the correct Vtable from **WPS/ungrib/Variable_Tables/** directory to the file name "Vtable" in the run directory. e.g.

```
ln -s ungrib/Variable_Tables/Vtable.GFS Vtable
```

- Link GRIB files using provided script **link_grib.csh**:

```
link_grib.csh /data/GRIB/gfs/gfs*
```



Running ungrib

- Type the following to run ungrib:

  `./ungrib.exe >& log.ungrib`

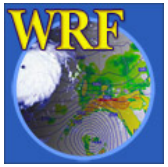
- If successful, you should see

  `Successful completion of ungrib`

- output files from ungrib, one per time period:

  `FILE:2009-04-15_00`

  `FILE:2009-04-15_06` …



Running metgrid

- Edit namelist records **`&share`** and **`&metgrid`**
- Type the following to run metgrid:

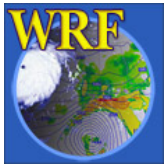
  `./metgrid.exe >& log.metgrid`
- If successful, you should see

  `Successful completion of metgrid`
- Output from metgrid program:

  `met_em.d01.2009-04-15_00:00:00`

  `met_em.d01.2009-04-15_06:00:00`

  `met_em.d02.2009-04-15_00:00:00` (for a nest, usually only one time period is needed)

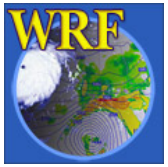


Typical Errors Running WPS

- Using wrong Vtable
- Missing some surface data, which may result an error message like:

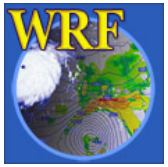
```
WRF_DEBUG: Warning DIM  4 , NAME
  num_metgrid_levels REDIFINED  by var
  TT  27 26  in wrf_io.F90 line 2420
  ERROR: Error in ext_pkg_write_field
```

- Missing soil temperature or moisture
- Check the log file from running ungrib to know what met fields you have got



Steps to Run real and wrf

1. cd to *run/* or one of the *test case* directories
2. Link or copy WPS output files to the directory for real-data cases
3. Edit *namelist.input* file for the appropriate grid and times of the case
4. Run initialization program (*ideal.exe, real.exe*)
5. Run model executable, *wrf.exe*



Running Real-Data Case

- If you have compiled the *em_real* case, you should have:

  real.exe - real data initialization program

  wrf.exe - model executable

  ndown.exe - program for doing one-way nesting

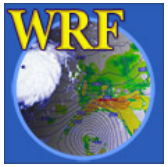
- These executables are linked to:

  WRFV3/run

  and

  WRFV3/test/*em_real*

➔ One can go to either directory to run.

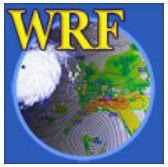


Running WRF Real-data Cases

- One must successfully run WPS, and create **`met_em.*`** file for more than one time period

- Link or copy WPS output files to the run directory:

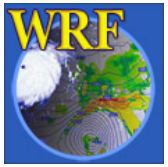
```
cd test/em_real
ln -s ../../../WPS/met_em.* .
```



Running WRF Real-data Cases

- Edit **`namelist.input`** file for runtime options (at mininum, one must edit **`&time_control`** for start, end and integration times, and **`&domains`** for grid dimensions)

Pay attention to first column in the **`namelist.input`** file if you are using a single domain. For nested runs, multiple columns of namelists need to be edited.



example of **namelist.input** file: **&time_control**

```
&time_control
run_days                            = 0,
run_hours                           = 24,
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2000, 2000, 2000,
start_month                         = 01,   01,   01,
start_day                           = 24,   24,   24,
start_hour                          = 12,   12,   12,
start_minute                        = 00,   00,   00,
start_second                        = 00,   00,   00,
end_year                            = 2000, 2000, 2000,
end_month                           = 01,   01,   01,
end_day                             = 25,   25,   25,
end_hour                            = 12,   12,   12,
end_minute                          = 00,   00,   00,
end_second                          = 00,   00,   00,
interval_seconds                    = 21600
history_interval                    = 180,  60,   60,
frame_per_outfile                   = 1000, 1000, 1000,
restart_interval                    = 360,
```

domain 1 option

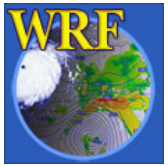
nest options

example of **namelist.input** file: **&domains**

```
&domains
time_step                           = 180
time_step_fract_num                 = 0,
time_step_fract_den                 = 1,
max_dom                             = 1,
e_we                                = 74,    112,    94,
e_sn                                = 61,    97,     91,
e_vert                              = 28,    28,     28,
num_metgrid_levels                  = 21
num_metgrid_soil_levels             = 4
dx                                  = 30000, 10000,  3333,
dy                                  = 30000, 10000,  3333,
eta_levels                          = 1.0,0.996,0.99,0.98,… 0.0
p_top_requested                     = 5000,
```

Match the dimensions defined in WPS

nest options



Running WRF Real-data Cases

Run the real-data initialization program:

`./real.exe`, if compiled serially / SMP, or

`mpirun -np N ./real.exe`, for a MPI job

where ***N*** is the number of processors requested.

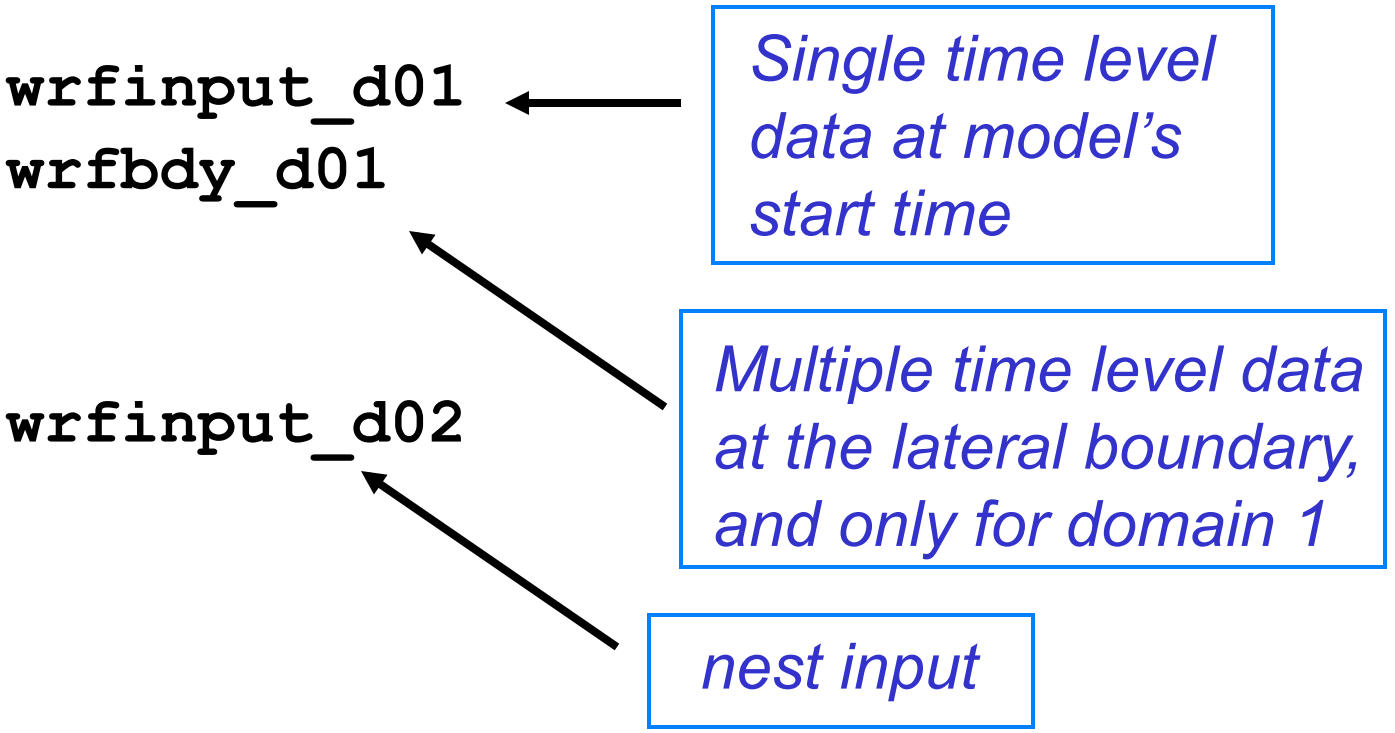
Depending on how the computer is configured, one may need to specify *machinefile* option:

```
mpirun -machinefile mach -np N ./real.exe
```

- here `'mach'` is a file containing a list of machine/ processor names

Running WRF Real-data Cases

- Successfully running this program will create model initial and boundary files:



Running WRF Real-data Cases

- Run the model executable by typing:

  `./wrf.exe >& wrf.out &`

  or

  `mpirun -np N ./wrf.exe &`

- Successfully running the model will a create model *history* file:

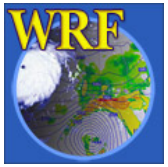
  `wrfout_d01_2005-08-28_00:00:00`

  `wrfout_d02_2005-08-28_00:00:00`

  And *restart* file if **`restart_interval`** is set to 720 :

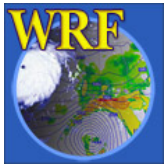
  `wrfrst_d01_2008-08-28_12:00:00`

  `wrfrst_d02_2008-08-28_12:00:00`



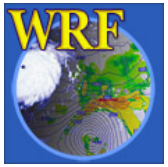
Where do I start?

- Always start with a *namelist* template provided in a test case directory, whether it is a ideal or real data case.
  - A number of namelist templates are provided in *test/test-case/* directories
- Use document to guide the modification of the namelist values:
  - run/README.namelist
  - User's Guide, Chapter 5
  - Full list of namelists and their default values can be found in Registry files: Registry.EM, and registry.io_boilerplate (IO options)



To run a job in a different directory..

- Directories *run/* and test_<*case*>/ are convenient places to run, but it does not have to be.
- Copy or link the content of these directories to another directory, including physics data files, wrf input and boundary files and wrf namelist and executables, and you should be able to run a job anywhere on your system.



Check Output

Output After a Model Run

- Standard out/error files:

  `wrf.out`, or **`rsl`**.* files

- Model history file(s):

  `wrfout_d01_<date>`

- Model restart file(s), optional

  `wrfrst_d01_<date>`

Output from a multi-processor run

The standard out and error will go to the following files for a MPI run:

`mpirun -np 4 .wrf.exe` ➔

`rsl.out.0000`	`rsl.error.0000`
`rsl.out.0001`	`rsl.error.0001`
`rsl.out.0002`	`rsl.error.0002`
`rsl.out.0003`	`rsl.error.0003`

There is one pair of files for each processor requested

What to Look for in a standard out File?

Check run log file by typing

`tail wrf.out`, or

`tail rsl.out.0000`

You should see the following if the job is successfully completed:

`wrf: SUCCESS COMPLETE WRF`

How to Check Model History File?

- Use **ncdump**:

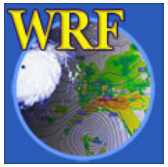
  ncdump -v Times wrfout_d01_<date>

  to check output times. Or

  ncdump -v U wrfout_d01_<date>

  to check a particular variable (U)

- Use **ncview** or **ncBrowse** (great tools!)
- Use post-processing tools (see talks later)



What is in a *wrf.out* or *rsl* file?

- A print of namelist options
- Time taken to compute one model step:

```
Timing for main: time 2000-01-24_12:03:00 on domain   1:    3.25000 elapsed seconds.
Timing for main: time 2000-01-24_12:06:00 on domain   1:    1.50000 elapsed seconds.
Timing for main: time 2000-01-24_12:09:00 on domain   1:    1.50000 elapsed seconds.
Timing for main: time 2000-01-24_12:12:00 on domain   1:    1.55000 elapsed seconds.
```

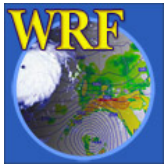
- Time taken to write history and restart file:

```
Timing for Writing wrfout_d01_2000-01-24_18:00:00 for domain   1:   0.14000 elapsed seconds.
```

- Any model error prints:

```
5 points exceeded cfl=2 in domain 1 at time 4.200000 MAX AT i,j,k: 123 48 3
     cfl,w,d(eta)= 4.165821
```

-> An indication the model has become numerically unstable



Simple Trouble Shooting

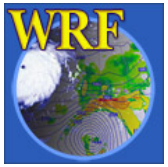
Often-seen runtime problems

- `module_configure: initial_config: error reading namelist: &dynamics`

  > Typos or erroneous namelist variables exist in namelist record &dynamics in *namelist.input* file

- `input_wrf.F: SIZE MISMATCH: namelist ide,jde,num_metgrid_levels= 70 61 27 ; input data ide,jde,num_metgrid_levels= 74 61 27`

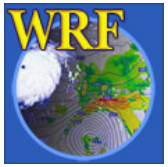
  > Grid dimensions in error



Often-seen runtime problems

- `Segmentation fault (core dumped)`
  - > Often typing **'unlimit'** or **'ulimit -s unlimited'** or equivalent can help when this happens quickly in a run.

- `121 points exceeded cfl=2 in domain 1 at time 4.200000 MAX AT i,j,k: 123 48 3 cfl,w,d(eta)= 4.165821`
  - > Model becomes unstable due to various reasons. If it happens soon after the start time, check input data, and/or reduce time step.



Resources

- Information on compiling and running WRF, and a more extensive list of namelist options and their definition / explanations can be found in the WRF User's Guide, Chapter 5
- namelist templates in test/em* directories
- example.namelists in test/em_real directory
- README.namelist in WRFV3/run
- Registry/Registry.EM