

Introduction

The WRF ARW model is a fully compressible, nonhydrostatic model (with a hydrostatic option). Its vertical coordinate is a terrain-following hydrostatic pressure coordinate. The grid staggering is the Arakawa C-grid. The model uses higher-order numerics. This includes the Runge-Kutta 2nd- and 3rd-order time integration schemes, and 2nd- to 6th-order advection schemes in both horizontal and vertical directions. It uses a time-split small step for acoustic and gravity-wave modes. The dynamics conserves scalar variables.

The WRF ARW model code contains initialization programs (ideal.exe and real.exe), a numerical integration program (wrf.exe), and a program to do one-way nesting (ndown.exe). The WRF ARW model Version 3 supports a variety of capabilities. These include:

- Real-data and idealized simulations
- Various lateral boundary condition options for both real-data and idealized simulations
- Full physics options
- Non-hydrostatic and hydrostatic (runtime option)
- One-way, two-way nesting and a moving nest
- Applications ranging from meters to thousands of kilometers

Software requirements

- Fortran 90 or 95 and C compiler
- perl 5.04 or later
- If MPI and OpenMP compilation is desired, MPI or OpenMP libraries are required
- WRF I/O API supports netCDF, pnetCDF, PHD5, GriB 1 and GriB 2 formats; hence one of these libraries needs to be available on the computer on which you compile and run WRF
- The following UNIX utilities: csh and Bourne shell, make, M4, sed, awk, and the uname command

Program Flow

- If you are only planning on running *Idealized* Cases, you would need:

WRF ARW Model + PostProcessing

- If you are planning on running *Real* Cases, you would need:

WPS + WRF ARW Model + PostProcessing

- If you are planning on running *Real Cases with Variational Analysis*, you would need:

WPS + WRF-Var + WRF ARW Model + PostProcessing

If you plan on using WRF-Var, be sure to reference [Chapter 6](#) in the WRF-ARW User's Guide, once you have become familiar with the basic WRF ARW Model.

View the [flowchart](#) to see how the programs interact

Documentation

[Users' Guide](#)

The OnLine Tutorial is complementary to the Users' Guide. The latest WRF and WPS code are described in this document. The Users' Guide is updated for every Bi-Annual Tutorial, and is, therefore, the most up-to-date guide for running the WRF ARW model. It is highly recommended to download this guide before you start running the model.

[WRF ARW Technical Note](#)

This document contains:

- Descriptions of ARW model equations, discretization, initialization, and nesting
- General description of physics options available in the model
- General description of the WRF-Var

[Bi-Annual Tutorial Presentations](#)

All the slide shows presented at the last Bi-Annual Tutorial Class

WRF-Var

Refer to [Chapter 6](#) in the WRF ARW User's Guide.

Browse the [WRF ARW Users Pages](#) for more information and documentation

Let's get Started

Program flow for real data cases:



BUT, the successful compilation of the WPS program **DEPENDS** on the successful compilation of the WRF ARW Model. The reason for this is that the two programs share common routines, like the WRF I/O API. Therefore, once we have the source code, we will first compile WRF ARW, and then WPS.

Program flow for idealized cases:



GET Source Code

The source code for the WRF ARW model is available from the download page:

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

NOTE: From the above page, go to *New Users* (users will be asked to register - it is free), or *Returning Users* (users must supply their email address before they can download the code again).



Also look at the "Known Problems and Fixes" pages for the latest information regarding bug fixes that have not yet been included in the released code:

[WRF Known Problems](#) ; [WPS Known Problems](#)

Download

- the latest WRF ARW tar file, and
- the latest WPS tar file (*if you plan on running real data cases*)

Working Directory

- Create a working directory **WRF/**
mkdir WRF
- Place the tar files you downloaded in your working directory

Unpack the Code

For Real Data cases:
gunzip WRFV3.TAR.gz
tar -xf WRFV3.TAR
gunzip WPS.TAR.gz
tar -xf WPS.TAR

This will create directories:

WRFV3/
WPS/

For Idealized Data cases:
gunzip WRFV3.TAR.gz
tar -xf WRFV3.TAR

This will create the directory:

WRFV3/

NOTE: The tar file names will contain the latest version number (not shown here)

Regardless of whether you are interested in Real or Idealized cases, WRF must be configured and compiled first, so let's examine the WRF code first.

Examine the WRF ARW Source Code

Move into the WRFV3 directory you created:

```
cd WRFV3
```

Inside this directory, you will find a number of files and directories.

Below are descriptions of some of the files and directories to take special note of.

- The README files contain useful information about the code and how to set up and run the model.

- Source code directories:

 - dyn_em/ Directory containing modules for dynamics in WRF ARW core

 - dyn_nmm/ Directory containing modules for dynamics in WRF NMM core

 - dyn_exp/ Directory for a 'toy' dynamical core

 - external/ Directory containing external packages, such as those for IO, time keeping and

 - MPI

 - frame/ Directory containing modules for WRF framework

 - inc/ Directory containing include files

 - main/ Directory for main routines, such as wrf.F, and all executables after install

 - phys/ Directory for all physics modules

 - share/ Directory containing mostly modules for WRF mediation layer and WRF I/O

 - tools/ Directory containing tools

- Scripts:

 - clean Script to clean created files, executables

 - compile Script for compiling WRF code

 - configure Script to configure the configure.wrf file for compile

- Makefile Top-level makefile

- Registry/ Directory for WRF Registry file

- arch/ Directory where compile options are gathered

- run/ Directory where one may run WRF

- test/ Directory that contains 7 test case directories, may be used to run WRF

Now that you have a feel for the WRF source code, you are ready to configure WRF for your computer.

Configure WRF

This step will create a configure file for your particular computer. The code has been ported to a wide variety of popular platforms.



Before you continue, ensure that your computer [environment](#) is set up correctly.

Now that you have set up your environment. Type:

```
./configure
```

You will be given a list of choices for your computer. These choices range from compiling for a single processor job, to using OpenMP shared-memory or distributed-memory parallelization options for multiple processors. Some options support nesting, others do not. So select the option carefully. For example, the choices for a Linux computer looks like this:

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /usr/local/netcdf-pgi
PHDF5 not set in environment. Will configure WRF for use without.
$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 (non-SGI installations) (serial)

12. Linux x86_64 i486 i586 i686, ifort compiler with icc (non-SGI installations) (smpar)
13. Linux x86_64 i486 i586 i686, ifort compiler with icc (non-SGI installations) (dmpar)
14. Linux x86_64 i486 i586 i686, ifort compiler with icc (non-SGI installations) (dm+sm)
15. Linux i486 i586 i686 x86_64, PathScale compiler with pathcc (serial)
16. Linux i486 i586 i686 x86_64, PathScale compiler with pathcc (dmpar)

Enter selection [1-16] :



Never compiled WRF before?

Pick a single-threaded option, as this will reduce the number of possible problems you run into. Once you are proficient in compile and running WRF, you can re-configure WRF for a more complicated environment.

NCAR - IBM Users

Option 3 is recommended on the NCAR IBM computers.

See [other available options](#).

For 2D idealized cases

Always pick a single threaded, no-nesting option



serial : single processor

smpar : shared memory option (OpenMP)

dmpar : distributed memory option (MPI)

dm+sm : Distributed Memory with Shared Memory (for example, MPI across nodes with OpenMP within a node) - usually better performance is through **dmpar** only

Once you have made this choice, you will be asked to choose the type of nesting run you are interested in

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



Most common option is "basic" (1)

3 is mainly used for moving nests following a hurricane vortex.

The above will create a [configure.wrf](#) file. Edit compile options/paths in this file, *if necessary*.



After typing ./configure, no options are listed OR the options are not for your platform

This will happen if the code has not been ported to your platform.
You will need to [add compilation options](#) for your computer.
(NOTE - this should be for advanced users)

Compiling for Real Data Cases



Compiling for Idealized Data Cases



Compile WRF for Real Data Cases



On some computers (*e.g. some Intel machines*), it is necessary to set the following environment variable before one compiles:

```
setenv WRF_EM_CORE 1
```

Type:

./compile

You will be given the following choices:

Usage:

compile [-d] [-j n] wrf compile wrf in run dir (NOTE: no real.exe, ndown.exe, or ideal.exe generated)

or choose a test case (see README_test_cases for details) :

compile em_b_wave

compile em_esmf_exp

compile em_fire

compile em_grav2d_x

compile em_heldsuarez

compile em_hill2d_x

compile em_les

compile em_quarter_ss

compile em_real

compile em_scm_xy

compile em_seabreeze2d_x

compile em_squall2d_x

compile em_squall2d_y

compile em_tropical_cyclone

```
compile exp_real
compile nmm_real
```

```
compile -d  compile without optimization and with debugging
compile -j n  parallel make using n tasks if supported (default 2)
compile -h  help message
```

Since we are compiling a WRF ARW real data case, type:

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

Check the [compile.log](#) file for any errors. If your compilation was successful, you should see the following executables created in the **main/** directory:

<code>main/ndown.exe</code>	Used for one-way nesting
<code>main/nup.exe</code>	<i>Upscaling - used with WRF-Var</i>
main/real.exe	WRF initialization for real data cases
main/wrf.exe	WRF model integration

These executables will be linked from the **main/** directory to the directories **run/** and **test/em_real/**, which is the directory from which you will be running the code.



The executables were not created

Check the [compile.log](#) file carefully for any errors.

Most common errors are:

- Incorrect netCDF version (*e.g., netCDF compiled with PGI, but you are compiling the code with Intel compilers*).
- Required libraries not installed, or not installed correctly.
- Paths to libraries not found.

Typically when you are unable to compile WRFV3, it is related to a problem with your environment, compiler(s), and/or libraries. We have a webpage that will walk you through the steps of verifying that your environment is set up properly, testing that each of the libraries are built correctly, and are compatible with your compilers. You can find that web page [here](#).

If you need to correct a problem in the **configure.wrf** file, be sure to

```
./clean -a
./configure
```


then make the change to **configure.wrf** before your attempt to compile the code again.



If you need to report the problem to wrfhelp

Repeat what you have done one more time:

```
./clean -a  
./configure  
./compile em_real >& compile.log
```

Then send the **compile.log** file, together with your computer/compiler information to wrfhelp

If the WRF has compiled correctly, you are ready to install WPS.

WRF Preprocessing System (WPS)

The next step for real data cases is to compile WPS.

You should not be here if you have not already successfully [compiled WRF](#).

Examine the WPS Source Code

Move into the **WPS** directory you created:

```
cd WPS
```

***NOTE:** If you are still in the **WRFV3/** directory, this should be one directory up from where you are now.*

Inside this directory, you will find a number of files and directories.

Below are descriptions of some of the files and directories to take special note of.

- The README files contain useful information about the code and how to set up and run the model.

- Source code directories:

geogrid/ Directory containing code to create the static data
metgrid/ Directory containing code to create input to WRFV3
ungrib/ Directory containing code to unpack GRIB data
util/ Directory containing some utilities

- Scripts:

clean Script to clean created files and executables
compile Script for compiling WPS code
configure Script to configure the 'configure.wps' file for compile
link_grib.csh Script to link GRIB files

- arch/ Directory where compile options are gathered
- namelist.wps WPS namelist, basic options
- namelist.wps-all_options WPS namelist, all available options

Environment Variable - NETCDF

Set the **NETCDF** environment variable.

*Typically, the netcdf libraries are located in **/usr/local/netcdf***

setenv NETCDF /usr/local/netcdf



Add this environment variable to your **.cshrc** file. This will ensure that it is always set correctly, and that you do not have to reset it every time you login.

LINUX Users

Ensure that your netcdf libraries were compiled with the same compiler which you are going to use to compile WRF ARW.

i.e., If you are compiling WRF ARW with a PGI compiler, your netcdf libraries must also be compiled with PGI.

And if you are compiling WRF ARW with an INTEL compiler, your netcdf libraries must also be compiled with INTEL.

NCAR - IBM Users

On NCAR's IBMs Bluefire, netCDF is installed in:
/usr/local/netcdf

Configure WPS

This step will create a configure file for your particular computer. The code has been ported to a wide variety of popular platforms.

Type:

./configure

You will be given a list of choices for your computer. These choices range from compiling for a single processor job, to using distributed-memory parallelization options for multiple processors. For example, the choices for a Linux computer look like this:

```
Will use NETCDF in dir: /usr/local/netcdf-pgi
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2
I/O...
```

Please select from among the following supported platforms.

1. PC Linux i486 i586 i686, PGI compiler serial, NO GRIB2
2. PC Linux i486 i586 i686, PGI compiler serial
3. PC Linux i486 i586 i686, PGI compiler DM parallel, NO GRIB2
4. PC Linux i486 i586 i686, PGI compiler DM parallel
5. PC Linux i486 i586 i686, Intel compiler serial, NO GRIB2
6. PC Linux i486 i586 i686, Intel compiler serial
7. PC Linux i486 i586 i686, Intel compiler DM parallel, NO GRIB2
8. PC Linux i486 i586 i686, Intel compiler DM parallel
9. PC Linux i486 i586 i686, g95 compiler, serial, NO GRIB2
10. PC Linux i486 i586 i686, g95 compiler, serial

Enter selection [1-10] :



NCAR - IBM Users

Please see [NCAR-IBM Options](#)

In General

If you don't have GRIB2 data, always pick a "NO GRIB2" option.



Have GRIB2 data?

You will need the following libraries: JasPer, libPNG and Zlib.
Download a tar file containing all 3 libraries [here](#).

Zlib may already be on your computer, so check before installing this library.
JasPer and PNG are compression libraries needed to ungrid GRIB version 2 data.

You will see a [configure.wps](#) file created. Edit compile options/paths in this file, *if necessary*.



WRF_DIR path

If your 'WRFV3' path is NOT `../WRFV3`, you should edit the [configure.wps](#) file and set the correct path to your 'WRFV3' directory.



After typing `./configure`, no options are listed OR the options are not for your platform
This will happen if the code has not been ported to your platform.
You will Compile WPS

Type:

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

Check the [compile.log](#) file for any errors. If your compilation is successful, you should see these executables created.

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

```
Generates static data  
Generates input data for WRFV3  
Unpacks GRIB data
```

A few utilities will also be *linked* under the **util/** directory.

avg_tsfc.exe	Computes daily mean surface temperature from intermediate files <i>Recommended for using with the 5-layer soil model (sf_surface_1 = 1) in WRF</i>
g1print.exe	Lists the contents of a GRIB1 file
g2print.exe	Lists the contents of a GRIB2 file
mod_levs.exe	Removes superfluous levels from 3-d fields in intermediate files
plotfmt.exe	Plots intermediate files (<i>dependent on NCAR Graphics - if you do not have these libraries, plotfmt.exe will not compile correctly</i>)

plotgrids.exe	Generates domain graphics- An excellent tool to configure domains before running geogrid.exe (<i>dependent on NCAR Graphics - if you don't have these libraries, plotgrids.exe will not compile correctly</i>)
rd_intermediate.exe	Reads intermediate files
calc_ecmwf_p.exe	Creates fields that have 3-d pressure and geopotential height on the same levels as the other atmospheric fields, for use with ECMWF sigma-level data sets
height_ukmo.exe	Computes a geopotential height field for data sets from the UKMO Unified Model



Utilities

Detailed explanations of the [WPS utility programs](#) are available from Chapter 3 of the User's Guide.



NCAR Graphics

If you don't have [NCAR Graphics](#) on your system, it is not a problem, but since it is a very handy, free plotting program, we recommend installing it at some point.



The executables were not created
Check the [compile.log](#) file carefully for any errors.

Most common errors are:

- Incorrect netCDF version (*e.g., netCDF compiled with PGI, but you are compiling the code with Intel compilers*).
- Paths to libraries not found.
- Required libraries not installed, or not installed correctly.
- You configured for GRIB2 data, but do not have the extra libraries (*JasPer, PNG and Zlib*) on your system, or they have been installed incorrectly. *To download a tar file containing all 3 libraries, click [here](#).* (*Zlib may already be on your computer, so check before installing this library. JasPer and PNG are compression libraries needed to ungrib GRIB version 2 data.*)

Typically when you are unable to compile WRFV3, it is related to a problem with your environment, compiler(s), and/or libraries. We have a webpage that will walk you through the steps of verifying that your environment is set up properly, testing that each of the libraries are built correctly, and are compatible with your compilers. You can find that web page [here](#).

If you correct any problems in the `configure.wps` file, be sure to

```
./clean
before your attempt to compile the code again.
```

You are now ready to run the WRF ARW model.

need to [add compilation options](#) for your computer.
(NOTE - this should be for advanced users)

Compile WPS

Type:

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

Check the [compile.log](#) file for any errors. If your compilation is successful, you should see these executables created.

geogrid.exe -> geogrid/src/geogrid.exe	Generates static data
metgrid.exe -> metgrid/src/metgrid.exe	Generates input data for WRFV3
ungrib.exe -> ungrib/src/ungrib.exe	Unpacks GRIB data

A few utilities will also be *linked* under the **util/** directory.

avg_tsfc.exe	Computes daily mean surface temperature from intermediate files. <i>Recommended for using with the 5-layer soil model (sf_surface_physics = 1) in WRF</i>
g1print.exe	Lists the contents of a GRIB1 file
g2print.exe	Lists the contents of a GRIB2 file
mod_levs.exe	Removes superfluous levels from 3-d fields in intermediate files
plotfmt.exe	Plots intermediate files (<i>dependent on NCAR Graphics - if you don't have these libraries, plotfmt.exe will not compile correctly</i>)
plotgrids.exe	Generates domain graphics- An excellent tool to configure domains before running geogrid.exe (<i>dependent on NCAR Graphics - if you don't have these libraries, plotgrids.exe will not compile correctly</i>)

rd_intermediate.exe	Reads intermediate files
calc_ecmwf_p.exe	Creates fields that have 3-d pressure and geopotential height on the same levels as the other atmospheric fields, for use with ECMWF sigma-level data sets
height_ukmo.exe	Computes a geopotential height field for data sets from the UKMO Unified Model



Utilities

Detailed explanations of the [WPS utility programs](#) are available from Chapter 3 of the User's Guide.



NCAR Graphics

If you don't have [NCAR Graphics](#) on your system, it is not a problem, but since it is a very handy, free plotting program, we recommend installing it at some point.



The executables were not created

Check the [compile.log](#) file carefully for any errors.

Most common errors are:

- Incorrect netCDF version (*e.g., netCDF compiled with PGI, but you are compiling the code with Intel compilers*).
- Paths to libraries not found.
- Required libraries not installed, or not installed correctly.
- You configured for GRIB2 data, but do not have the extra libraries (*JasPer, PNG and Zlib*) on your system, or they have been installed incorrectly. *To download a tar file containing all 3 libraries, click [here](#).* (*Zlib may already be on your computer, so check before installing this library. JasPer and PNG are compression libraries needed to ungrib GRIB version 2 data.*)

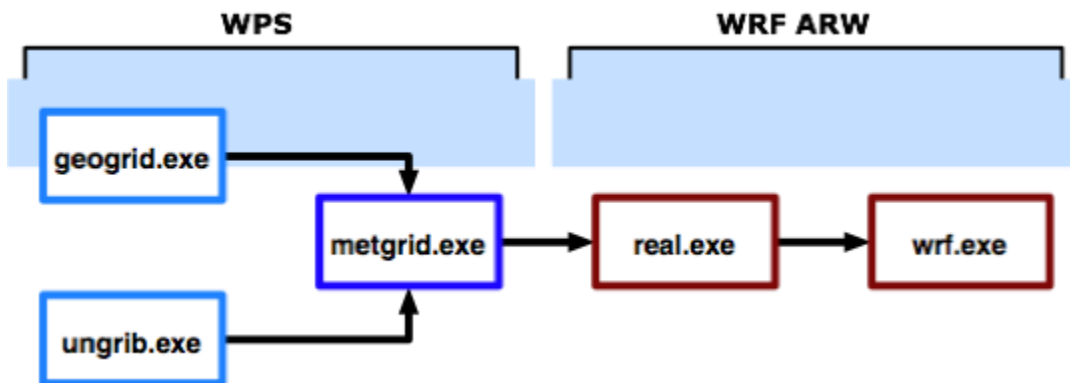
Typically when you are unable to compile WRFV3, it is related to a problem with your environment, compiler(s), and/or libraries. We have a webpage that will walk you through the steps of verifying that your environment is set up properly, testing that each of the libraries are built correctly, and are compatible with your compilers. You can find that web page [here](#).

If you correct any problems in the configure.wps file, be sure to
./clean
before your attempt to compile the code again.

You are now ready to run the WRF ARW model.

Basics for Running the Model

Below is a depiction of the program flow during a typical model run.



WPS

geogrid.exe creates terrestrial data (*static*).

ungrib.exe unpacks GRIB meteorological data and packs it into an [intermediate](#) file format.

metgrid.exe interpolates the meteorological data horizontally onto your model domain.

Output from **metgrid.exe** is used as input to WRF.

WRF ARW

real.exe vertically interpolates the data onto the model coordinates.

wrf.exe generates the model forecast.



Detailed information on all components are available in the [User's Guide](#). Peruse Chapter 3 for details on WPS, and Chapter 5 for details on WRF ARW.



Frequently Asked Questions

Should I run GEOGRID or UNGRIB first?

It does not matter in which order these two programs are run, as they are independent from each other. *Additionally, re-running one does not require that you re-run the other.*

Depending on your case setup you may choose to run one or the other first.

Before we run a specific case, let's first look at some details regarding each of the individual components.

Basics - GEOGRID

The purpose of **GEOGRID** is to define the simulation domain(s), and interpolate various terrestrial data sets to the model grids.

The simulation domain(s) are defined using information specified by the user in the "**share**" and "**geogrid**" sections of the [WPS namelist](#).

By default, in addition to computing latitude and longitude for every grid point, geogrid will interpolate soil categories, land use category, terrain height, annual mean deep-soil temperature, monthly vegetation fraction, monthly albedo, maximum snow albedo, and slope category to the model grids.

STEPS to Run GEOGRID

1. [Download](#) the terrestrial input data
2. Edit [namelist.wps](#)



Run **./util/plotgrids.exe** to ensure your domain is in the right location before running **geogrid.exe**.

After running **./util/plotgrids.exe**, type the following to view your domain: `idt`
`gmeta`

3. Run **geogrid.exe** (output is 'geo_em.dxx.nc' - *one file for each domain*)

Basics - UNGRIB

The purpose of **UNGRIB** is to unpack **GRIB** (*GRIB1 and GRIB2*) meteorological data and pack it into an [intermediate file format](#).

Unpacking the data is controlled via the "**share**" and "**ungrib**" sections of the [WPS namelist](#).

UNGRIB

- is NOT dependent on any WRF model domain.
- is NOT dependent on GEOGRID.
- does NOT cut down the data according to your model domain specification. It simply unpacks the required fields and writes them out into a format that the **METGRID** program can read.
- makes use of **Vtables** (see sample Vtables in the **WPS/ungrib/Variable_Tables/** directory) to specify which fields to unpack from the GRIB files. The **Vtables** lists the fields and their GRIB codes that must be unpacked from the GRIB files.



Although Vtables are provided for many common data sets, it would be impossible for ungrib to anticipate every possible source of meteorological data in GRIB format. When a new source of data is to be processed by UNGRIB, the user may create a new Vtable, either from scratch, or by using an existing Vtable as an example. In either case, a basic knowledge of the meaning and use of the various fields in the Vtable will be helpful.

Required Fields

When creating your own Vtables, note that a number of input fields are required.

STEPS to Run UNGRIB

1. [Download](#) data and place in a [unique directory](#)
2. [Familiarize yourself with the data](#)
3. **Link** (with the UNIX command *ln*) the correct Vtable
-For example, if you are using GFS data, type:
ln -sf ungrib/Variable_Tables/Vtable.GFS Vtable
4. **Link** (with supplied script [link_grib.csh](#)) the input GRIB data
./link_grib.csh *path_to_data*
5. Edit [namelist.wps](#). You only need to pay attention to the following parameters:
start_date ; end_date ; interval_seconds ; prefix
Note: Normally one will leave the "prefix" set to "FILE", except in cases where this may overwrite data.
6. **Run** ungrib.exe (*output will be intermediate files - one file for each time*)
7. [Familiarize yourself with the intermediate files](#)



WRF will need boundary conditions for the **ENTIRE** time you plan on running the

model. So make sure to UNGRIB enough input data.



Input data NOT in GRIB format?

Write a stand alone program to read your data and output it in the [intermediate](#) file format. (This "intermediate" link has detailed file format information and sample programs. Extra information is also available in [Chapter 3](#) of the User's Guide.)

Creating your intermediate files manually eliminates the need for the UNGRIB step. If you create your intermediate files manually, do check the correctness of your intermediate files with the utilities programs `rd_intermediate.exe` and `plotfmt.exe` (available in the `WPS/util` directory).

Input data in netCDF format.

[Learn](#) how to write FORTRAN programs that will read and write netCDF files.



Frequently Asked Questions

Do I have to re-run ungrib if I use the same data, but change my domain location?

No. UNGRIB is not tied to any domain, but if you are using a regional model as input data make sure that your new domain will still fit inside the input data domain.

Do I have to re-run ungrib if I add a nest to my model run?

No. UNGRIB has no WRF model domain information.

For which length of time do I need to run ungrib?

As WRF is a regional model, you need input data for the entire time you plan on running the WRF model.

Basics - METGRID

The purpose of **METGRID** is to horizontally interpolate the meteorological data onto your model domain.

Output from `metgrid.exe` is used as input to WRF.

The simulation domain(s) are defined using information specified by the user in the "**share**" and "**metgrid**" sections of the [WPS namelist](#).

STEPS to Run METGRID

1. Edit [namelist.wps](#).
2. Input to METGRID is the **geo_em.dxx.nc** output files from GEOGRID, and the intermediate output files from UNGRIB.
3. Run **metgrid.exe**

Output from this run will be:

- met_em.d01.YYYY-MM-DD_DD:00:00.nc - one file for per time, and
- met_em.dxx.YYYY-MM-DD_DD:00:00.nc - one file for per nest, for the initial time only
(*met_em files for other times can be created, but are only needed for special FDDA runs*).

Basics - WRF

The WRF model has 2 steps:

real.exe

This program vertically interpolates the **met_em*** files (*generated by metgrid.exe*), creates boundary and initial condition files, and does some consistency checks.

wrf.exe

Generates the model forecast.

STEPS to Run WRF

1. **Move** to the directory in which you plan to run the code (*either WRFV3/test/em_real or WRFV3/run*).
 - Both the [directories](#) **run/** and **test/em_real/** will have all the files you need linked in, so it does not matter which one you choose to be in to run the code.
2. **Link or copy** (with the UNIX command **ln / cp**) the **met_em** files to this directory.
`ln -sf path_to_met_em_files/met_em.d0*`
3. Edit [namelist.input](#).
4. **Run real.exe** ([verify](#) that the program runs correctly)
 - You should have the following **output files** (*default setup*): **wrfinput_d01** & **wrfbdy_d01**
 - This is true for single domain and default nested runs.
 - If **wrfinput_dxx** files are required for nested domains, these can be created
(*more on this in 'nested case studies'*).
5. **Run wrf.exe** ([verify](#) that the program runs correctly)
 - You should have the following **output files** (*default setup*): **wrfout_dxx_[initial_date]** (*one for each domain*)
 - Each file (*by default*) will contain all the forecast output times.

6. Use a [post-processing tool](#) to view the output.



Does one need to re-run `real.exe` if a change to the `namelist.input` has been made?

-Generally, one only needs to run **wrf.exe** again.

-If you have changed the LSM packages you are running with, then you must re-run **real.exe** again.

-However, **real.exe** runs very fast - so if in doubt, just go ahead and run **real.exe**.

Let's run a real case study

January 2000 Case

Set up the Model domain (`geogrid.exe`)

Ensure that that you have the [terrestrial](#) data somewhere on your computer

1. Ensure you are in the **WPS/** directory

2. Edit **namelist.wps**

Make the following 2 changes:

```
max_dom = 1
geog_data_path = 'Your WPS_GEOG data location'
```

3. Check that the rest of the domain settings look as follows:

Note 1: *You should not need to make any changes, as these are the default settings*

Note 2: *As we are only interested in domain 1, you can ignore all values in the second column*

Note 3: *For explanation of these variables see [Chapter 3](#) of the User's Guide, or [namelist.wps](#)*

```
parent_id = 1,
parent_grid_ratio = 1,
i_parent_start = 1,
j_parent_start = 1,
e_we = 74,
e_sn = 61,
geog_data_res = '10m',
dx = 30000,
```

```
dy = 30000,  
map_proj = 'lambert',  
ref_lat = 34.83,  
ref_lon = -81.03,  
truelat1 = 30.0,  
truelat2 = 60.0,  
stand_lon = -98.0,
```

4. Before running *geogrid.exe*, let's make sure the domain is in the right location (you need *NCAR Graphics* installed on your system to use this utility - if you don't have [NCAR Graphics](#) and you don't want to install it now, you may skip this step, and continue to step 5 below).

```
./util/plotgrids.exe
```

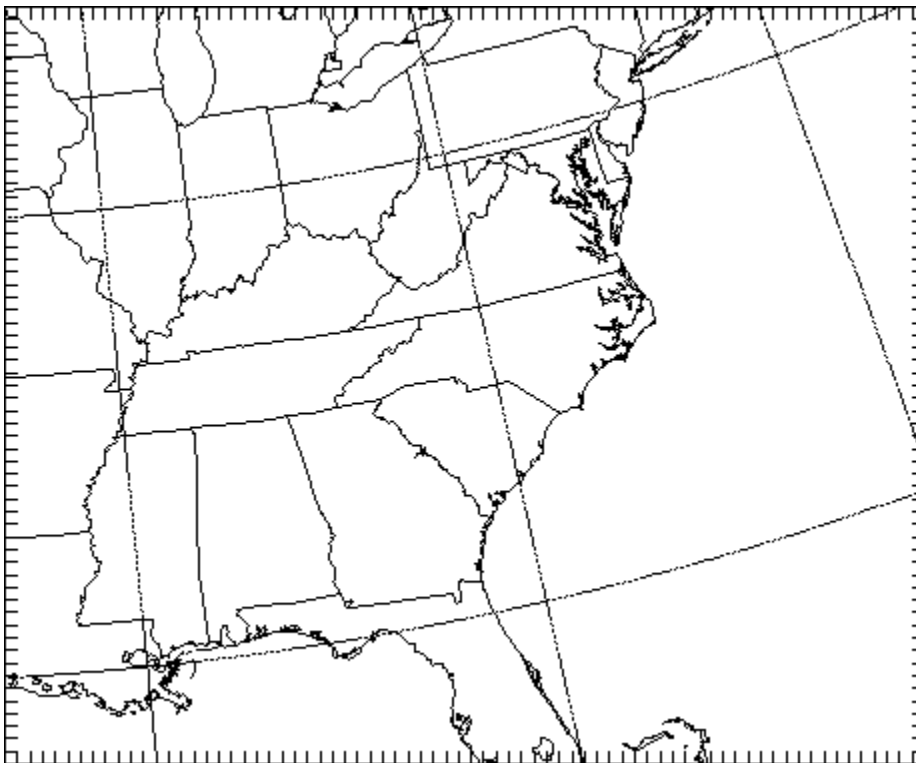
This will create the following file:

```
gmeta
```

To view the domain you are about to create, type:

```
idt gmeta
```

You should see the following domain:



5. Now run *geogrid.exe* to create static data for this domain:

```
./geogrid.exe
```

The text below should appear on your screen during run time. Look for "*Successful completion of geogrid*" at the end of the run, indicating that all went well during the execution. A [geogrid.log](#) file, containing more information, will also be created.

```
Parsed 11 entries in GEOGRID.TBL
Processing domain 1 of 1
Processing XLAT and XLONG
Processing MAPFAC
Processing F and E
Processing ROTANG
Processing LANDUSEF
Calculating landmask from LANDUSEF (WATER = 16)
Processing HGT_M
Processing HGT_U
Processing HGT_V
Processing SOILTEMP
Processing SOILCTOP
Processing SOILCAT
Processing SOILCBOT
Processing ALBEDO12M
Processing GREENFRAC
Processing SNOALB
Processing SLOPECAT
Processing SLOPECAT
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Successful completion of geogrid.  !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Make sure that the static file has been created:

```
geo_em.d01.nc
```

6. Check the contents of this file:

- a. Use the `ncdump` utility:
[ncdump -h geo_em.d01.nc](#)
- b. Use the [ncview](#) utility.
- c. This file can also be viewed with any of the supported [graphical](#) tools.



Below is the output file generated by this step:
[geo_em.d01.nc](#)

January 2000 Case

Unpack input GRIB data (ungrib.exe)

1. **Download** the Tutorial case study [data](#) and place in directory ../DATA/JAN00

The data we are going to use as input data are the operational NCEP Eta/NAM 212 grid (40km) model analyses (**AWIP**). These are the same as the NCEP NAM model data, but contain only analysis.

For information about the **AWIP** archives at NCAR: See, <http://dss.ucar.edu/datasets/ds609.2/>.

Type: GRIB1 data

Resolution:

Area - approx. 60W–140W ; 20N–60N ; at 40 km

[Analysis times](#) - 00Z and 12Z

26 pressure levels (1000-50hPa ; excluding surface)

The data are available for the period 2000-01-24_12 to 2000-01-26_00 (data frequency is 6 hourly).

The 'tm06' files have been renamed to the corresponding dates.

2. Let's get [familiar](#) with the data by running **g1print** on one of the data files:

[./util/g1print.exe ../DATA/JAN00/2000012412.AWIPSF](#)

3. Link in the AWIP Vtable:

In -sf ungrib/Variable_Tables/**Vtable.AWIP** Vtable

Peruse the ungrib/Variable_Tables/Vtable.AWIP file to see which fields we are going to try and unpack from the GRIB files.

4. Link-in the GRIB data by making use of the script [link_grib.csh](#)

[./link_grib.csh ../DATA/JAN00/2000012](#)

This will create the following links:

```
GRIBFILE.AAA -> ../DATA/JAN00/2000012412.AWIP3D
GRIBFILE.AAB -> ../DATA/JAN00/2000012412.AWIPSF
GRIBFILE.AAC -> ../DATA/JAN00/2000012418.AWIP3D
GRIBFILE.AAD -> ../DATA/JAN00/2000012418.AWIPSF
GRIBFILE.AAE -> ../DATA/JAN00/2000012500.AWIP3D
GRIBFILE.AAF -> ../DATA/JAN00/2000012500.AWIPSF
GRIBFILE.AAG -> ../DATA/JAN00/2000012506.AWIP3D
GRIBFILE.AAH -> ../DATA/JAN00/2000012506.AWIPSF
GRIBFILE.AAI -> ../DATA/JAN00/2000012512.AWIP3D
GRIBFILE.AAJ -> ../DATA/JAN00/2000012512.AWIPSF
GRIBFILE.AAK -> ../DATA/JAN00/2000012518.AWIP3D
GRIBFILE.AAL -> ../DATA/JAN00/2000012518.AWIPSF
GRIBFILE.AAM -> ../DATA/JAN00/2000012600.AWIP3D
GRIBFILE.AAN -> ../DATA/JAN00/2000012600.AWIPSF
```

5. **Edit** *namelist.wps*, and set the following:

Note: For explanation of these variables see [Chapter 3](#) of the User's Guide, or [namelist.wps](#)

```
start_date = '2000-01-24_12:00:00',
end_date = '2000-01-25_12:00:00',
interval_seconds = 21600,
prefix = 'FILE',
```

6. **Run** `ungrib` to create the [intermediate](#) files:

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

This will create the following files:

```
FILE:2000-01-24_12
FILE:2000-01-24_18
FILE:2000-01-25_00
FILE:2000-01-25_06
FILE:2000-01-25_12
```

The log file ([ungrib_data.log](#)) created during this step is very important. It contains information regarding the fields that were found in the input file, and on which levels these fields are available.

Peruse this log to make sure all the fields in the Vtable were found.

7. **Familiarize** yourself with the intermediate files:

[./util/rd_intermediate.exe FILE:2000-01-24_12](#)
[./util/plotfmt.exe FILE:2000-01-24_12 ; idt gmeta](#)



[AWIP data](#) for other time periods is available from NCAR/DSS



Below are the output intermediate files generated by this step.
If you have problems generating this data, you can use this set to continue to the next step.

[Intermediate files](#)

January 2000 Case

Interpolate the input data onto our model domain (metgrid.exe)

1. **namelist.wps**

The *namelist.wps* file should be set up correctly for the default case.

Note: For explanation of the *namelist* variables see [Chapter 3](#) of the *User's Guide*, or [namelist.wps](#)

2. **Run** *metgrid.exe* to interpolate the input data on our model domain:

```
./metgrid.exe
```

The text below should appear on your screen during run time. Look for "*Successful completion of metgrid*" at the end of the run, indicating that all went well during the execution. A [metgrid.log](#) file containing more information will also be created.

```
Processing domain 1 of 1
Processing 2000-01-24_12
FILE
Processing 2000-01-24_18
FILE
Processing 2000-01-25_00
FILE
Processing 2000-01-25_06
FILE
Processing 2000-01-25_12
FILE
```

!!
! Successful completion of metgrid. !
!!

This step will create the following files :

```
met_em.d01.2000-01-24_12:00:00.nc
met_em.d01.2000-01-24_18:00:00.nc
met_em.d01.2000-01-25_00:00:00.nc
met_em.d01.2000-01-25_06:00:00.nc
met_em.d01.2000-01-25_12:00:00.nc
```

3. Check the contents of this file:

- a. Use the ncdump utility:
ncdump -h met_em.d01.2000-01-24_12:00:00.nc
- b. Use the [ncview](#) utility.
- c. This file can also be viewed with any of the supported [graphical](#) tools.



Below are the output *met_em.d01* files generated by this step.
If you have problems generating this data, you can use this set to continue to the next step.
[met_em.d01 files](#)

January 2000 Case

Run the model (real.exe & wrf.exe)

1. Ensure you are in the **WRFV3/** directory. For this case we are going to run in the **test/em_real/** directory.

```
cd test/em_real
```

2. **Link-in** the *met_em* files created with *metgrid.exe*

```
ln -sf ../../WPS/met_em.d01.2000-01* .
```

3. Edit [namelist.input](#)

Note 1: *You should not need to make any changes, as these are the default settings, but check to ensure they are set as below*

Note 2: *As we are only interested in domain 1, you can ignore all values in other columns*

Note 3: For explanation of these variables see [Chapter 5](#) of the User's Guide, or [namelist.input](#)

```
run_days = 0,  
run_hours = 12,  
run_minutes = 0,  
run_seconds = 0,  
start_year = 2000,  
start_month = 01,  
start_day = 24,  
start_hour = 12,  
end_year = 2000,  
end_month = 01,  
end_day = 25,  
end_hour = 12,  
interval_seconds = 21600  
input_from_file = .true.,  
history_interval = 180,  
frames_per_outfile = 1000,  
time_step = 180,  
max_dom = 1,  
s_we = 1,  
e_we = 74,  
s_sn = 1,  
e_sn = 61,  
s_vert = 1,  
e_vert = 28,  
num_metgrid_levels = 27  
dx = 30000,  
dy = 30000,
```

4. **Run** real.exe (*verify that the program runs correctly*)
Check that the following two files have been created:

```
wrfinput_d01  
wrfbdy_d01
```

What is [contained](#) in these files?



```
wrfinput_d01  
wrfbdy_d01
```

5. **Run** wrf.exe (*verify that the program runs correctly*)
Check that the following file has been created

wrfout_d01_2000-01-24_12:00:00

This file will contain all the time periods for this run.



[wrfout](#)

6. Quick check of the contents of this file:

a. Use the ncdump utility:

```
ncdump -h wrfout_d01_2000-01-24_12:00:00
```

```
ncdump -v Times wrfout_d01_2000-01-24_12:00:00 (to see which forecast times are in the file)
```

b. Use the [ncview](#) utility.

7. Use any of the supported [graphical](#) tools to display the model output.

OnLine Tutorial Case Studies

[Default January 2000 case](#)

[Single Model Domain case](#)

[Restart Run](#)

You must do the Single Model Domain case before you attempt a restart.

[Input SST into model](#)

[Nested Model Runs](#)

[Analysis nudging](#)

Physics:

Pick any of the above case studies and change the physical options

Review the namelist options in [Chapter 5](#) of the User's Guide or [namelist.input](#)

