WRF Installation

**Note :** Red color lines are commands which you need to type in terminal

Before installing WRF check the compilers what you have

1. Gfortran/ pgf90
2. Gcc
3. M4

**How to check**

1. Which gfortran
2. Which gcc (also check gcc version and it should be 4.0 or more )
3. Which m4

Then you will their locations of compilers.

Then create two new directories called **Build\_WRF & TESTS**

**Then go to TESTS folder and download this below link**

wget <http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/Fortran_C_tests.tar>

Then you’ll have one tar file and extract the file with below command

tar -xf Fortran\_C\_tests.tar

And after test the 7 which you got after extraction, find below the commands

**Tets1**#

gfortran TEST\_1\_fortran\_only\_fixed.f

./a.out

**Tets2**#

gfortran TEST\_2\_fortran\_only\_free.f90

./a.out

**Tets3**#

gcc TEST\_3\_c\_only.c

./a.out

**Tets4**#

gcc -c -m64 TEST\_4\_fortran+c\_c.c

gfortran -c -m64 TEST\_4\_fortran+c\_f.f90

gfortran -m64 TEST\_4\_fortran+c\_f.o TEST\_4\_fortran+c\_c.o

./a.out

**Tets5**#

./TEST\_csh.csh

**Tets6**#

./TEST\_perl.pl

**Tets7**#

./TEST\_sh.sh

**Then after go to Build\_WRF directory and build the libraries**

Create a directory for libraries

mkdir LIBRARIES

Download the 5 tar files which dependies for WRF

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/mpich-3.0.4.tar.gz>

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/netcdf-4.1.3.tar.gz>

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/jasper-1.900.1.tar.gz>

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/libpng-1.2.50.tar.gz>

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/zlib-1.2.7.tar.gz>

**It is important to note that these libraries must all be installed with the same compilers as will be used to install WRF and WPS.**

**Modify the .bashrc file in the home directory of current user to set the environment variable.**

**nano ~/.bashrc.**

**At the bottom of the .bashrc file add the lines so that they wills et for future logins**

**# WRF environment variables**

**export DIR=/delta/home/jeevan/WRF/build\_WRF/libraries**

**export CC=gcc**

**export CXX=g++**

**export FC=gfortran**

**export CFLAGS=-m64**

**export F77=gfortran**

**export FFLAGS=-m64**

**Then source the .bashrc file to make these settings active for current session**

**source ~/.baschrc**

**follow these steps in order**

1. Then start extract the netCDF tar file

tar xzvf netcdf-4.1.3.tar.gz

cd netcdf-4.1.3

./configure --prefix=$DIR/netcdf --disable-dap --disable-netcdf-4 --disable-shared

make  
make install

Then open .bashrc file and modify and set two new variables at the bottom. And the source file to make these settings active for current session.

export PATH=$DIR/netcdf/bin:$PATH

export NETCDF=$DIR/netcdf

Then source to bash

source ~/.bashrc

1. Then configuring the **MPICH** library and If you have single core processor you may skip it

tar xzvf mpich-3.0.4.tar.gz     #or just .tar if no .gz present  
cd mpich-3.0.4  
./configure --prefix=$DIR/mpich  
make  
make install

1. **zlib**: This is a compression library necessary for compiling WPS (specifically ungrib) with GRIB2 capability.

Assuming all the "setenv" commands from the NetCDF install are already set, you can move on to the commands to install zlib.

go to bashrc script and export the below paths

export LDFLAGS=-L$DIR/grib2/lib

export CPPFLAGS=-I$DIR/grib2/include

save & exit

tar xzvf zlib-1.2.7.tar.gz     *#or just .tar if no .gz present*  
cd zlib-1.2.7  
./configure --prefix=$DIR/grib2  
make  
make install  
cd ..

1. **libpng**: This is a compression library necessary for compiling WPS (specifically ungrib) with GRIB2 capability.

Assuming all the "export" commands from the NetCDF install are already set, you can move on to the commands to install zlib.

tar xzvf libpng-1.2.50.tar.gz     *#or just .tar if no .gz present*  
cd libpng-1.2.50  
./configure --prefix=$DIR/grib2  
make  
make install  
cd ..

1. **JasPer**: This is a compression library necessary for compiling WPS (specifically ungrib) with GRIB2 capability  
   Assuming all the "setenv" commands from the NetCDF install are already set, you can move on to the commands to install zlib.

tar xzvf jasper-1.900.1.tar.gz     *#or just .tar if no .gz present*  
cd jasper-1.900.1  
./configure --prefix=$DIR/grib2  
make  
make install  
cd ..

**Libraries compatibility test**

Once the target machine is able to make small Fortran and C executables (what was verified in the System Environment Tests section), and after the NetCDF and MPI libraries are constructed (two of the libraries from the Building Libraries section), to emulate the WRF code's behavior, two additional small tests are required. We need to verify that the libraries are able to work with the compilers that are to be used for the WPS and WRF builds.

Go to TESTS directory and download below link

<http://www2.mmm.ucar.edu/wrf/OnLineTutorial/compile_tutorial/tar_files/Fortran_C_NETCDF_MPI_tests.tar>

* To unpack the tar file, type:

tar -xf Fortran\_C\_NETCDF\_MPI\_tests.tar

* There are 2 tests:
  1. **Test #1**: Fortran + C + NetCDF  
       
     The NetCDF-only test requires the include file from the NETCDF package be in this directory. Copy the file here:

cp ${NETCDF}/include/netcdf.inc .

Compile the Fortran and C codes for the purpose of this test (the -c option says to not try to build an executable). Type the following commands:

gfortran -c 01\_fortran+c+netcdf\_f.f  
gcc -c 01\_fortran+c+netcdf\_c.c  
gfortran 01\_fortran+c+netcdf\_f.o 01\_fortran+c+netcdf\_c.o -L${NETCDF}/lib -lnetcdff -lnetcdf  
./a.out

The following should be displayed on your screen:

C function called by Fortran  
Values are xx = 2.00 and ii = 1  
SUCCESS test 1 fortran + c + netcdf

* 1. **Test #2**: Fortran + C + NetCDF + MPI

The NetCDF+MPI test requires include files from both of these packages be in this directory, but the MPI scripts automatically make the mpif.h file available without assistance, so no need to copy that one. Copy the NetCDF include file here:

cp ${NETCDF}/include/netcdf.inc .

Note that the MPI executables mpif90 and mpicc are used below when compiling. Issue the following commands:

mpif90 -c 02\_fortran+c+netcdf+mpi\_f.f  
mpicc -c 02\_fortran+c+netcdf+mpi\_c.c  
mpif90 02\_fortran+c+netcdf+mpi\_f.o 02\_fortran+c+netcdf+mpi\_c.o -L${NETCDF}/lib -lnetcdff -lnetcdf  
mpirun ./a.out

The following should be displayed on your screen:

C function called by Fortran  
Values are xx = 2.00 and ii = 1  
status = 2  
SUCCESS test 2 fortran + c + netcdf + mpi

Building WRF

Dowloand the WRF version and extract the tar file

Then go to WRf folder and

./configure

Then choose number 34 because of damper (distributed memory)

And then compile with case name em\_les (in my case)

**em** stands for ‘Eulerian mass-coordinate’ solver

./compile *em\_les* >& log.compile

Then check the .exe files in main folder

ls -ls main/\*.exe

Building WPS

Download the WPS source code(below link) into build\_WRF directory

<http://www2.mmm.ucar.edu/wrf/src/WPSV4.0.TAR.gz>

Then extract the tar file

cd WPS

./clean

Then you need to configure the WPS but before you need to set some paths then you can configure

nano ~/.bashrc

go to last line and edit these lines

export JASPERLIB=$DIR/grib2/lib  
  
export JASPERINC= $DIR/grib2/include

then save and exit

source ~/.bashrc

./configure

Then choose 1

the metgrid.exe and geogrid.exe programs rely on the WRF model's I/O libraries. There is a line in the configure.wps file that directs the WPS build system to the location of the I/O libraries from the WRF model:

go to nano configure.wps file and check below text

WRF\_DIR = ../WRF\_les

You can now compile WPS:

./compile >& log.compile

If the compilation is successful, there should be 3 executables in the WPS top-level directory, that are linked to their corresponding src/ directories:

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

## \*geogrid.exe will create the domine

Now you need Geogrid file to create domain file

🡺then download GEOG-high resolution file

wget <https://www2.mmm.ucar.edu/wrf/src/wps_files/geog_high_res_mandatory.tar.gz>

🡺once it downloads then extract the tar file

tar xzf geog\_high\_res\_mandatory.tar.gz

after extraction it will create **WPS\_GEOG** folder contains all domain input folders

🡺 namelist.wps file is input for pre-processing and it look like below

&share

wrf\_core = 'ARW',

max\_dom = 2, ===🡺 it will tell how many nested domains you need

start\_date = '2019-09-04\_12:00:00','2019-09-04\_12:00:00',

end\_date = '2019-09-06\_00:00:00','2019-09-04\_12:00:00',

interval\_seconds = 10800

/

&geogrid

parent\_id = 1, 1,

parent\_grid\_ratio = 1, 3,

i\_parent\_start = 1, 53,

j\_parent\_start = 1, 25,

e\_we = 150, 220, =====<<>>> points of direction

e\_sn = 130, 214,

geog\_data\_res = 'default','default',

dx = 15000, ====🡺 resolution in meters

dy = 15000,

map\_proj = 'lambert',

ref\_lat = 33.00, =====🡺>>> center of domain latitude

ref\_lon = -79.00, =====🡺>>> center of domain longitude

truelat1 = 30.0,

truelat2 = 60.0,

stand\_lon = -79.0,

geog\_data\_path = '/glade/work/wrfhelp/WPS\_GEOG/' ======<>>> this path should be where WPS\_geog directory (in my case /delta/home/jeevan/WRF/build\_WRF/wrf\_real/WPS\_GEOG)

/

&ungrib

out\_format = 'WPS',

prefix = 'FILE',

/

&metgrid

fg\_name = 'FILE'

/

🡺After editing namelist.wps then run the ./geogrid.exe to check the running.

You can see geo\_em.d01.nc files (is domain we created)

You can view & explore this file with ncview or panapoly(you must install in local mechine)

🡺 then I created GFS directory and downloaded gfs files

Then you should download the gfs files from NECP or ECMWF (your choice )

I downloaded (below link) from NCEP for trail run

<https://nomads.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/gfs.20210608/00/atmos/>

then in the WPS directory there is file “link\_grib.csh” and ungrib folder.

In the ungrib folder you can see several vtables and in those vtables will help to give meteorological input.

🡺 after you downloaded grib files to **GFS** directory go to **WPS** directory and link gfs files to WPS

./link\_grib.csh /delta/home/jeevan/WRF/build\_WRF/wrf\_real/GFS

./ungrib.exe =====<<>> this will ungrib the gfs files

🡺 after you need to link metgrid/METGRID.TBL.ARW to WPS directory as MATGRID.TBL. this is enough to start metgrid

./metgrid.exe ====<<>> this will run met paramters

After that Link those met files to **run** directory in **WRF** directory

🡺 then here we go to main job

We go to main directory in WRF, there you can see ./real.exe & ./wrf.exe

./real.exe = it will create initial and boundary conditions

./wrf.exe = it will integrate all setup to model and simulate atmosphere development with respect to time.

🡺 our job will start from **run** directory mostly

🡺 namelist is main configuration file in our WRF model.

Once you modify the name list run the real executable.

./real.exe or (mpirun -n 10 ./real.exe ====<>>> this mpirun will help to number of processors to finish task sooner)

It will create wrfinput\_d01 wrfbdy\_d01 input files

wrfbdy\_d01 = this means initial boundary conditions

wrfinput\_d01= initial atmospheric conditions

🡺 then you can run the ./wrf.exe same way

./wrf.exe

Then here comes the post processing of WRF output. You can use any data analysis software.