Compiling wgrib2 v2.0.6+

Compiling wgrib2 is easy on a linux system with gcc/gfortran or Windows with the cygwin compilers

```
1) Download ftp://ftp.cpc.ncep.noaa.gov/wd51we/wgrib2/wgrib2.tgz
2) remove pre-existing grib2 directory if exists: rm -r grib2
3) untar wgrib2.tgz:  tar -xzvf wgrib2.tgz    (use gnu tar)
4) cd to main directory:  cd grib2
5) define the C and fortran compilers and make
   Bash:
     export CC=gcc
     export FC=gfortran
     make
     make lib                          only if you want the ftn_api
   Csh
     setenv CC gcc
     setenv FC gfortran
     make
     make lib                          only if you want the ftn_api
6) See if wgrib2 was compiled
     wgrib2/wgrib2 -config

note: you may have to install gcc and gfortran
```

Compiling with the intel compilers on a linux system is supported with v2.0.6. However, some code still needs to be compiled with gcc, so icc and gcc both need to called from the same makefile. I wouldn't be surprised that some systems are configured so that you can load a GNU module in order to use to use gcc or you can load an INTEL module in order to use icc/ifort but you can't load both modules at the same time. In this case, the default makefile will not work. The work around is to disable jpeg200 support.

```
1) Download ftp://ftp.cpc.ncep.noaa.gov/wd51we/wgrib2/wgrib2.tgz
2) remove pre-existing grib2 directory if exists: rm -r grib2
3) untar wgrib2.tgz:  tar -xzvf wgrib2.tgz    (use gnu tar)
4) cd to main directory:  cd grib2
5) define the C and fortran compilers and make
   Bash:
     export CC=icc
     export FC=ifort
     export COMP_SYS=intel_linux
     make
     make lib                          only if you want the ftn_api
   Csh
     setenv CC icc
     setenv FC ifort
     setenv COMP_SYS intel_linux
     make
     make lib                          only if you want the ftn_api

6) See if wgrib2 was compiled
     wgrib2/wgrib2 -config
```

Compiling with Clang

```
1) download ftp://ftp.cpc.ncep.noaa.gov/wd51we/wgrib2/wgrib2.tgz
2) remove pre-existing grib2 directory if exists: rm -r grib2
3) untar wgrib2.tgz:  tar -xzvf wgrib2.tgz    (use gnu tar)
4) cd to main directory:  cd grib2
5) If your version of Clang does not support OpenMP, edit the makefile
   old: USE_OPENMP=1
   new: USE_OPENMP=0
```

```
6) define the C and fortran compilers and make
   Bash:
     export CC=clang
     export FC=gfortran -fplugin=dragonegg
     make
     make lib                          only if you want the ftn_api
   Csh
     setenv CC clang
     setenv FC "gfortran -fplugin=dragonegg"
     make
     make lib                          only if you want the ftn_api
7) See if wgrib2 was compiled
     wgrib2/wgrib2 -config
```

The default wgrib2 configuration requires both a C and f90+ compiler. In order combine C and fortran code, the C and Fortran compilers need to be compatible with each other. The supported pairs of compilers are (gcc, gfortran), (gcc,g95), (icc, ifort), (opencc, openf95), and (xlc, xlf).

If there is no supported fortran compiler, wgrib2 can be compiled with only a C compiler.

```
1) download ftp://ftp.cpc.ncep.noaa.gov/wd51we/wgrib2/wgrib2.tgz
2) remove pre-existing grib2 directory if exists: rm -r grib2
3) untar wgrib2.tgz:  tar -xzvf wgrib2.tgz    (use gnu tar)
4) cd to main directory:  cd grib2
5) edit the makefile and change
     USE_IPOLATES=1 -> USE_IPOLATES=0
     MAKE_FTN_API=1 -> MAKE_FTL_API=0
6) define the CC environement variable and make
   Bash:
     export CC=(your C compiler)
     make
   Csh
     setenv CC (your C compiler)
     make
7) See if wgrib2 was compiled
     wgrib2/wgrib2 -config
```

Compiling wgrib2 on OS X. See Grace Peng's blog. Compiling wgrib2 on OS X. See Daniel Bowman's blog

Comments:

1. For Cygwin, use the linux instructions
2. For Windows, use Cygwin compilers and bash shell and the linux instructions
3. For Windows 10, you can use WSL and the Ubuntu compilers. Use the generic linux instructions
4. Using g2clib is only recommended for developers of g2clib
5. Intel and the Portland compilers are not compatible with the Jasper library

Netcdf4

Wgrib2 can be compiled with netcdf4 instead of netcdf3. The advantages of netcdf4 include compression and support for larger files. The cost is that the wgrib2 executable is about 3 times larger for an option that you might not use. Consequently the default configuration is to support creation of netcdf3 files. For people who need netcdf4 support, you can convert the netcdf3 file into netcdf4 or recompile wgrib2 to directly produce netcdf4. To do this, you you need to change the makefile from,

```
Default makefile:
..
USE_NETCDF3=1
USE_NETCDF4=0
..

Modified makefile for netcdf4
..
USE_NETCDF3=0
USE_NETCDF4=1
..
```

When you run make, it will fail twice with requests that you download the netcdf4 and hdf5 source codes. After you download the two software packages, you can make wgrib2.

Missing zlib.h

I haven't been able to reproduce this compiling problem. The png compression requires both the zlib library and the zlib.h include. I suspect that the libpng configure program is using the system zlib because it finds the system zlib library. Perhaps the fix is to install the zlib.h files by installing the "zlib-develop" package. Another fix is to remove the system zlib and let libpng use the wgrib2-supplied zlib. The third fix is to diable png compression in the make file. PNG compression is not very good and I have only seen it used by RADAR files from ESRL.

wgrib2 compile questions

```
Question: Why do you build japser with the following flags?

     --disable-libjpeg --disable-opengl

Answer:

Libjpeg is not needed by wgrib2.  The makefile should work when
the system doesn't have libjpeg installed.

Opengl is not needed by wgrib2. Makefile should still work when the
system doesn't have opengl installed.

The wgrib2 makefile compiles the libraries in a manner optimized for wgrib2.
The libraries are not meant for general use.  For example, the netcdf library
doesn't have a fortran interface because I want the makefile to
work if a user lacks a fortran compiler.

Question: why do you include zlib and libpng?

Answer:  Not all linux distributions include libpng or they give it a different name.
Some linux distributions may have different versions of zlib installed.

Question: Why don't you make netcdf4 the default netcdf package?

Answer:

The Netcdf4/hdf5 libraries are very big and the end result is a long
download time and a large executable. If you decide to compile with
Netcdf4, the make will fail with the commands needed to download
the netcdf4 and hdf5 libraries.  After you have download the libraries,
the make should work.

Question: I get the error messagee: /bin/sh: ./configure: Permission denied

Answer:

It is possible to configure a filesystem so that the execute bit is
set to zero. Programs and scripts will not execute when they reside
on that filesystem.  Try compiling on another filesystem.  If that doesn't
work, you are probably not allowed to download programs.  The solution
is to ask that your sys admin to compile the program for you.  But if
your sys admin will compile code for you, why are you reading this page?
Let him read this page.

Question: Why the options to turn off aec, jpeg2000 and png compression

Answer:

Some HPC machines use one type of cpu and/or OS to compile and
another type of cpu and/or OS to run (cross-compile).  Using configure
scripts can be a challenge when cross-compiling.  I wanted a reasonable
wgrib2-subset that can be cross-compiled.  Consequently all libraries that
that require a configure script are optional.

Question: Why does compiling wgrib2 with the Netcdf option fail?

Answer:

This frequently happens with 32-bit machines.  This problem goes away
on a 64-bit platform or if netcdf-4 is used instead of netcdf-3. This
problem is not being addressed because of a lack of 32-bit machines.

Question: Why the option to turn off g2clib?

Answer: g2clib is not needed but it can be installed for testing.
```

Question: Why gctpc and proj4? Why not proj4 without gctpc?

Answer: Gctpc and proj4 are both projection libraries and only one is really needed. Proj4 has more functionality, is modern and has an active support group. Gctpc is old and is more-or-less unsupported. On the other hand, gctpc is simpler, works and supports OpenMP like the rest of wgrib2. Gctpc supports a different threading model. As grids get larger, you want the speed of gctpc. Both libraries are being supported in order to help debugging and in the case that a future grid requires Proj4 support.

Question: Why so many compile options?

Answer:
Some systems do not support certain POSIX features
Some functionality is not needed by most systems.
not all systems have fortran, MySQL, supported NetCDF/HDF
The code that calls IPOLATES is not using a standard interface
Some systems do not handle cross-compiling easily

Problem: I compiled wgrib2 and when I run it, it complains about missing libraries.

Answer:
I have the same problem with our Cray. If you compile with the Intel compilers, you have to load the Intel environment before running the executables. If you compile with the GNU compilers, you have to load the GNU environment before running. The way to get around this problem is to compile wgrib2 with the same compiler that was used to compile the system commands. (An old version of gcc/gfortran.)

Question: I am compiling with Intel Compilers and undefined references.

Answer:
If you get an error message to a "ompc" routine like

```
Ncpu.c:(.text+0x24): undefined reference to `ompc_set_num_threads'
```

This is because you are using an old Intel compiler. The makefile uses the current option to enable OpenMP rather than the old option.

NOAA/ National Weather Service
National Centers for Environmental Prediction
Climate Prediction Center
5830 University Research Court
College Park, Maryland 20740
Climate Prediction Center Web Team

Page last modified: Dec 29, 2017