wgrib2: changes since 5/2012

Operational wgrib2 on NCEP’s WCOSS is 5/2012 version

Brief look at the changes since then

Wesley Ebisuzaki   1/2015
Changes: grib table

Grib table has been updated

Codes will fail if they use the generic name such as

--- short generic names ---
varM_N_O
  M=discipline
  N=parameter category
  O=parameter number

--- long generic names ---
var discipline=0 master_table=2 parmcat=1 parm=190
var discipline=0 center=99 local_table=1 parmcat=1 parm=195
Changes: g2clib optional

wgrib2 used g2clib for decoding grib.
   Fedora and Redhat versions of wgrib2 were failing because
       they were using the opn g2clib without my patches. (Policy)
g2clib fails when finds unknown template.
g2clib is not parallelized

<table>
<thead>
<tr>
<th></th>
<th>OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>-g2clib=0</td>
<td>internal</td>
<td>internal</td>
</tr>
<tr>
<td>-g2clib=1</td>
<td>g2clib</td>
<td>internal with CF bug</td>
</tr>
<tr>
<td>-g2clib=2</td>
<td>N/A</td>
<td>g2clib (optional)</td>
</tr>
</tbody>
</table>

g2clib has CF bug, decimal scaling is ignored with constant field

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>g2clib is now optional for general distribution, may be include with NCEP distributions</td>
</tr>
</tbody>
</table>
Changes: Geolocation

Geolocation: determine the latitude and longitudes of the grid points. Need to go from spherical earth to ellipsoidal.

Internal code: donated+some iplib-based, spherical
gctpc USGS, orphaned? ellipsoidal
Proj4 Open source, gold standard ellipsoidal, uses config, OpenMP?

<table>
<thead>
<tr>
<th>OLD</th>
<th>NEW</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>internal (spherical)</td>
</tr>
<tr>
<td>-gctpc 1</td>
<td>gctpc</td>
</tr>
<tr>
<td>-proj4 1 (new)</td>
<td>N/A</td>
</tr>
</tbody>
</table>

* transition in progress, reasons: ellipsoidal, no config, OpenMP internal codes still used for Gaussian, rotated lat-lon
Changes: extended names

Names: want unique names to identify the fields

Old: HGT, TMP, PRES, UGRD

New:

Aerosols are called MASSMR (mass mixing ratio)
Need to know type and size
":TMP:2 m above ground:" is not TMP2m when error field

Names → extended names
MASSMR.aerosol=Dust_Dry.aerosol_size_>=2e-06,<3.6e-06.
TMP.analysis/forecast_error

-set_ext_name 1
Upgraded functionality: internal

-new_grid: e-grid, Mercator
(many): support for ensembles
(many): support for more tables
(many): bug fixes

Callable wgrib2 (v2.0.2)
  Development project with outside user
  C code can call wgrib2
  Can transfer data by arguments
  Only need to open file once
  Can be used for read and writing grib
  New
Upgraded functionality: external

<table>
<thead>
<tr>
<th>Script/Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>grb1to2.pl</td>
<td>grib1 to grib2 converter, perl script that calls wgrib and wgrib2. Easy to build?</td>
</tr>
<tr>
<td>fast_grib2_mean.sh</td>
<td>many times faster than previous script</td>
</tr>
<tr>
<td>wgrrib2m</td>
<td>script to parallelize wgrib2 by running multiple copies of wgrib2</td>
</tr>
<tr>
<td>g2grb.gs</td>
<td>GrADS script to write grib2 files</td>
</tr>
<tr>
<td>netcdf, HDF -&gt; grib2</td>
<td></td>
</tr>
<tr>
<td>alt_g2ctl</td>
<td>alternative g2ctl/gribmap (GrADS)</td>
</tr>
<tr>
<td>rNOMADS</td>
<td>package for R for reading NCEP grib, by Daniel Bowman</td>
</tr>
<tr>
<td>grib-filter/g2subset</td>
<td>user can download regridded fields</td>
</tr>
<tr>
<td>fort_grib_wrt</td>
<td>fortran callable routines to write grib2</td>
</tr>
</tbody>
</table>
fort_grib_wrt:

Fortran callable routines to write grib. Based on having an existing grib2 file (template). Templates are made by regridding existing grib files (wgrib2) and modifying metadata (wgrib2).

Step 1: open a channel (file number)
Step 2a: create metadata string (based on wgrib2 inventory)
   ex. ‘d=2015010212:HGT:500 mb:12 hour fcst:’
Step 2b: write grib message
Step 3: close channel
Fort_grib_wrt: Step 1

call grb_wopen(ichannel, file, template_file, grid_size, n_templates, compress, options, ierr)

integer :: ichannel
character (len=*) :: file
character (len=*) :: template_file

integer :: grid_size(n_templates)
integer :: n_templates
character (len=*) :: compress
character (len=*) :: options
integer :: ierr

textual explanation:
- **ichannel**: integer 1..4, channel
- **file**: name of the output file
- **template_file**: name of grib2 file with one or more templates
- **grid_size**: grid size for each template
- **n_templates**: num. of templates
- **compress**: compression: c1, c2, c3, j
- **options**: wgrib2 options (see doc)
- **ierr**: returns 0 if no errors
call grb_write(ichannel, template, data, metadata, ierr)

integer :: ichannel  integer
integer :: template   template number
real :: data(ndata)  real, values of the grid.
default: WE:SN, WE:NS (see doc)
undefined value = 9.999e20

character (len=*) :: metadata  metadata line
integer :: ierr    returns 0 if no errors

metadata is really simple .. wgrib2 inventory
'd=2015010100:HGT:500 mb:6 hour fcst:
call grb_wrt(1,1,data,'d=2015012100:TMP:200 mb:12 hour fcst', ierr)
Close channel

call grb_close(ichannel, ierr)

integer :: ichannel          integer, channel
Integer :: ierr              integer, 0 for no errors
ss2gg (sigma spectral to Gaussian grid) was modified to write grib2 using these routines. Modifications were painless and ss2gg is now a light-weight post processor.

To handle more esoteric fields like aerosols would require updates to the -set_metadata option in wgr2ib.
call system("grep "match string" file.inv | wgrib2 -i file.grb -bin file.bin")
open(unit=10,file="file.bin" ....)
read(10) grid
close(10)

Not high performance .. but ok for climate, models like GFS
Reading grib from fortran (future)

Callable wgrib2
Should be possible to write a higher performance fortran reading API using callable wgrib2.
Operational issues

Interpolation: testing as NAM post-processor
   Attractions: Can select type of interpolation
       -net_grid_interpolation bilinear \n       -if ":(SOTYP|VEGTYP)::" -new_grid_interpolation neighbor -fi

   Attraction: Can run faster (multiple processors by wgrib2m)

Complex packing with bitmap: NAM
   Added (v2.0.2)
   Complex packing with bitmaps is inferior because it produces
       larger files and is slower to read (wgrib2)
   Compatible with some codes
Operational issues: updated wgrib2

Parts of NCO have said that versioning of utilities is a good idea and needs approval within NCO. They wanted a bundle of new utilities with the new hardware.

For now, need to include version of wgrib2 with package that you want to installed.