

OpenIFS workshop 2019: Using ecCodes to work with GRIB files

Using the GRIB tools

The GRIB tools are part of the ECMWF ecCodes Library. They are a set of command line programs for interactive and batch decoding and processing of GRIB data

They provide ready and tested solutions to the most common processing of GRIB data and work with both GRIB edition 1 and GRIB edition 2.

All of the tools use a common syntax:

```
grib_<tool> [options] grib_file grib_file ... [output_grib]
```

There are tools for getting information about the ecCodes installation and the keys available

- `grib_info`, `grib_keys`

There are tools to inspect the content of and compare GRIB messages

- `grib_ls`, `grib_dump`, `grib_get`, `grib_get_data`, `grib_compare`

There are tools for counting and copying some messages

- `grib_count`, `grib_copy`

There is a tool for making changes to the content of a GRIB message

- `grib_set`

There is an advanced tool with a macro language provided via a rules file which allows for more sophisticated manipulation of GRIB files

- `grib_filter`

There is a tool for converting from GRIB to netCDF

- `grib_to_netcdf`

Getting help with the tools

Running any of the ecCodes tools without options or input files will produce a UNIX 'man page' style output describing the usage. Try:

```
getting help
```

```
grib_ls
```

Listing the contents of GRIB files

An inventory of the contents of GRIB files can be obtained with the **`grib_ls`** command.

1. For a single file, try:

```
grib_ls ICMGGepc8+000000
```

2. Multiple files:

```
grib_ls ICMGG*
```

3. Counting the number of messages in a file

```
grib_count ICMGGepc8+000000
```

Customising the output of grib_ls

1. Specifying the information to display:

```
grib_ls -p dataDate,dataTime,stepRange,shortName,typeOfLevel,level ICMGGepc8+000000
```

2. Specifying the type of key (s: string, i: integer). To output the centre as a string (default):

```
grib_ls -p centre:s ICMGGepc8+000000
```

or as an integer:

```
grib_ls -p centre:i ICMGGepc8+000000
```

3. Listing specific **namespace** information. Try:

```
grib_ls -n parameter ICMGGepc8+000000
```

and

```
grib_ls -n statistics ICMGGepc8+000000
```

Other namespaces to try are **time**, **vertical** and **geography**.

4. Order the output. Try:

```
grib_ls -B "level desc" ICMSHepc8+000000
```

Compare and contrast with:

```
grib_ls -B "level:i desc" ICMSHepc8+000000
```

5. The **where** option: **IS**

```
grib_ls -w shortName=st11 ICMGGe pc8+000000
```

IS NOT

```
grib_ls -w shortName!=st11 ICMGGe pc8+000000
```

OR

```
grib_ls -w shortName=st11/z ICMGGe pc8+000000
```

AND

```
grib_ls -w shortName=q,level=10 ICMGGe pc8+000000
```

6. Specifying the step unit. Compare the output from:

```
grib_ls -n time ICMGGe pc8+000000
```

and

```
grib_ls -s stepUnits=m -n time ICMGGe pc8+000012
```

The **stepUnits** can be set to **s, m, h, 3h, 6h, 12h, D, M, Y, 10Y, 30Y** or **C**. The result must be expressible as an integer (try setting **stepUnits=D** above). This is a temporary change for display purposes only !

⚠ **grib_ls** is a tool meant mainly for interactive use. If you want to use the tools in a shell script then take a look at **grib_get** instead. It works in much the same way as **grib_ls** but is less verbose and also fails if an error occurs.

Obtaining detailed information about a GRIB message with grib_dump

1. To print all of the keys available for a particular GRIB message:

```
grib_dump -w shortName=st11 ICMGGe pc8+000012
```

2. To obtain a listing which is more like the WMO tables:

```
grib_dump -0 -w shortName=st11 ICMGGe pc8+000012
```

3. To obtain even more detail:

```
grib_dump -0atH -w shortName=st11 ICMGGe pc8+000012
```

Printing the data values

For grid point data, the **`grib_get_data`** tool prints a list of latitude, longitude and data values from one or more GRIB files.

1. Try


```
grib_get_data -w shortName=z ICMGGePC8+000000
```

2. The format of the data values can be controlled with a C-style format statement. Try:

```
grib_get_data -w shortName=z -F "%.4f" ICMGGePC8+000000
```

3. Other information can be printed along with the lat-lon-values triplet:

```
grib_get_data -p dataDate,dataTime,stepRange -w shortName=st11 -F "%.4f" ICMGGePC8+000000
```


 If you need the values in a program then do not be tempted to use `grib_get_data` to output the information to a file and then read the file in your program. It is more efficient to use the appropriate Fortran 90, C or Python interface to ecCodes to access the data values directly.

Using the 'smart' copy

The structure of GRIB allows files to be simply copied and concatenated with the UNIX 'cat' command. However, it's sometimes desirable to do something more sophisticated. The **`grib_copy`** tool provides a way to copy only selected content from GRIB files.

1. To copy all of the pressure level data at 1000 hPa from a number of separate files and store the output in a single file, try:

```
grib_copy -w level=1000 ICMSHePC8* lev1000.grib
```

 Remember that you can check the output file with **`grib_ls`** to see the result !

2. To do the same with only temperature field and to store in descending forecast step order, try:

```
grib_copy -w level=1000,shortName=t -B "stepRange:i desc" ICMSHePC8* lev1000_ordered.grib
```

3. Now let's gather all of the temperature fields and store in order of decreasing level and decreasing step:

```
grib_copy -w shortName=t -B "level:i desc, stepRange:i desc" ICMSHePC8* all_t_ordered.grib
```

⚠ It is important when using the "order by" (-B) option to specify the key type with, e.g., "level:i" to avoid unexpected (unwanted ?) results. Remember that the key values are strings by default. Usually, we want a numeric order.

4. Key names can be used to specify the output file names. Try:

```
grib_copy ICMGGepc8* "[shortName]_[level].grib[edition]"
```

This provides a very convenient way to filter GRIB messages into separate files.

Setting key/value pairs

The **grib_set** tool allows key values to be set or modified. It is also able to make simple global changes to the data values themselves.

1. Suppose we want to use the soil temperature on level 1 to provide a sea-surface temperature. We can change the header information for this parameter and set it to sea-surface temperature

```
grib_set -S -s shortName=sst -w shortName=stl1 ICMGGepc8+* sst.grib
```

⚠ By default, **all** parameters are copied to the output file. The 'strict' option (-S) writes only those messages that are changed to the output file. To see the difference, try running `grib_set` again but without the -S option !

2. This is still not encoded like a true sea-surface temperature. To do this, we need also to change the `typeOfLevel`:

```
grib_set -S -s shortName=sst,typeOfLevel=surface -w shortName=stl1 ICMGGepc8+* sst.grib
```

3. An offset can be added to all data values:

```
grib_set -S -s offsetValuesBy=-273.15 -w shortName=stl1 ICMGGepc8+* stl1_in_degC.grib
```

⚠ Remember: use "`grib_ls -n statistics`" to make a quick check of the values in output file.

4. The values can be multiplied by a constant factor:

```
grib_set -S -s scaleValuesBy=0.102 -w shortName=z ICMGGepc8+* gh.grib
```


5. Some values are read only and cannot be changed. Try:

```
grib_set -S -s scaleValuesBy=0.000001,units="kg cm** -2" -w shortName=tcw ICMGGepc8+000000
tcm_kgcm-2.grib
```

The units key cannot be set as it is read only. the only way to change the units key is by changing the parameter (either the shortName or paramId), assuming there is a suitable parameter defined to express Total column water in units of kg/cm².


- All data values can be set to a constant

```
grib_set -S -d 1.0 -w shortName=tcw ICMGGepec8+000000 out.grib
```

 The resulting file is small as it contains only the reference value. This is the method used to create the ifs_sample files used to provide 'template' GRIBs for model output.

- Key names can be used to specify the output file names. This works in the same way as for **grib_copy**. Try:

```
grib_set -S -s scaleValuesBy=0.102,shortName=gh -w shortName=z ICMGGepec8+*
"[shortName]_[dataDate]_[dataTime]_[stepRange].grib"
```

-  **grib_set** is very powerful but it **cannot** be used to:
- make transformations to the data representation
 - transform data from one grid representation to another
 - select sub-areas of data

Resources and further reading

- The WMO FM 92 GRIB Manuals can be obtained from www.wmo.int/pages/prog/www/WMOCodes.html
- The ECMWF GRIB API manual is available at <https://software.ecmwf.int/wiki/display/GRIB/Home>
- ECMWF parameter database: <http://apps.ecmwf.int/codes/grib/param-db/>
- GRIB 1 keys <http://apps.ecmwf.int/codes/grib/format/grib1/>
- GRIB 2 keys <http://apps.ecmwf.int/codes/grib/format/grib2/>
- Edition independent keys <http://apps.ecmwf.int/codes/grib/format/edition-independent/>
- MARS specific keys <http://apps.ecmwf.int/codes/grib/format/mars/>
- The GRIB API Tools are documented at [GRIB tools](#)
- Details of the GRIB API Fortran 90 interface: [Fortran package grib_api](#)
- Example Python scripts and Fortran 90 and C programs can be found at [GRIB API examples](#)
- ECMWF GRIB API training course material: [GRIB API: Library and Tools](#)