# Using GRIB Tools

## Computer User Training Course 2019

**Paul Dando & Cristian Simarro & Xavi Abellan**

**User Support**

**servicedesk@ecmwf.int**

# ecCodes documentation

- **The ecCodes documentation and support pages are available at**

  https://confluence.ecmwf.int/display/ECC/ecCodes+Home

- **The GRIB Tools are documented at**

  https://confluence.ecmwf.int/display/ECC/GRIB+tools

  **Includes some examples of how to use the tools**

- **The ecCodes software can be downloaded from**

  https://confluence.ecmwf.int/display/ECC/Releases

# ecCodes keys and parameters for GRIB – THE Reference

- Parameters in GRIB
  - GRIB Parameter Database - https://apps.ecmwf.int/codes/grib/param-db

- ecCodes GRIB keys - https://apps.ecmwf.int/codes/grib/
  - GRIB Edition 1 - https://apps.ecmwf.int/codes/grib/format/grib1/
  - GRIB Edition 2 - https://apps.ecmwf.int/codes/grib/format/grib2/
  - GRIB Edition Independent - https://apps.ecmwf.int/codes/grib/format/edition-independent/

## Disclaimer

*The official copy of the FM-92 GRIB document from which the relevant information contained in above pages is derived can be obtained from the WMO web site: http://www.wmo.int/pages/prog/www/WMOCodes.html*

# GRIB Tools Quiz

- **grib_ls/grib_dump**

# Challenges

- Work in your $SCRATCH

```
cd $SCRATCH
```

- Make a copy of the challenges directory in your $SCRATCH

```
tar -xvf /home/ectrain/trx/ecCodes/grib_tools.tar
```

- This will create a directory in your $SCRATCH containing the GRIB data files for all the GRIB tools challenges
- There is a sub-directory for each practical:

```
ls $SCRATCH/grib_tools

challenge1 challenge2 challenge3 challenge4
```

# Challenge 1: inspecting GRIB messages using grib_dump

1. Experiment with using the different grib_dump options (`-O`, `-a` and `-t`). Inspect the GRIB message in the files file1.grib1 and file1.grib2 and identify:
   - the GRIB edition used to encode the messages
   - the (MARS)parameter ID, date, time, forecast step and the grid geometry
   - What are the maximum, minimum and average values of the fields?

# Challenge 1: inspecting GRIB messages with grib_ls

2. Use grib_ls to print the centre, dataDate, stepRange, levelType, shortName and paramId for msl.grib1 and msl.grib2 and order by ascending stepRange
   - Experiment with both –P and –p options and ‘key:i’, ‘key:s’
   - Which keys does grib_ls show by default for the two files ? What fields do they contain
3. Find the value of the MSLP at the grid point nearest to ECMWF (Lat 51.42°N, Lon 0.95° W) at each forecast step
   - What is the lat-lon value of the grid point nearest to ECMWF ?
   - How far is the chosen grid point from ECMWF ?
   - Use the file lsm.grib1 to provide a land-sea mask - are all four nearest grid points land points (mask ≥ 0.5) ?

# Generic ecCodes tools

- There is a tool for getting information about the ecCodes installation
  - codes_info
- There is a tool for counting GRIB or BUFR messages
  - codes_count
- There is a tool to split an input file (GRIB, BUFR etc) into chunks of roughly the same size
  - codes_split_file
- There is a GUI tool to inspect the content of a GRIB or BUFR file
  - codes_ui

# codes_info – information about ecCodes installation

**The generic codes_info tool gives basic information about the ecCodes package being used**

- ecCodes Version
- Path to definition files: ECCODES_DEFINITION_PATH
- Path to sample files: ECCODES_SAMPLES_PATH

```
> codes_info
ecCodes Version 2.10.0

Default definition files path is used:
   /usr/local/apps/eccodes/2.10.0/GNU/6.3.0/share/eccodes/definitions
Definition files path can be changed setting ECCODES_DEFINITION_PATH environment variable

Default SAMPLES path is used:
   /usr/local/apps/eccodes/2.10.0/GNU/6.3.0/share/eccodes/samples
SAMPLES path can be changed setting ECCODES_SAMPLES_PATH environment variable
```

# codes_count – count GRIB or BUFR messages

- **Counts (very quickly) the number of GRIB or BUFR messages in a list of files**
- **Syntax**

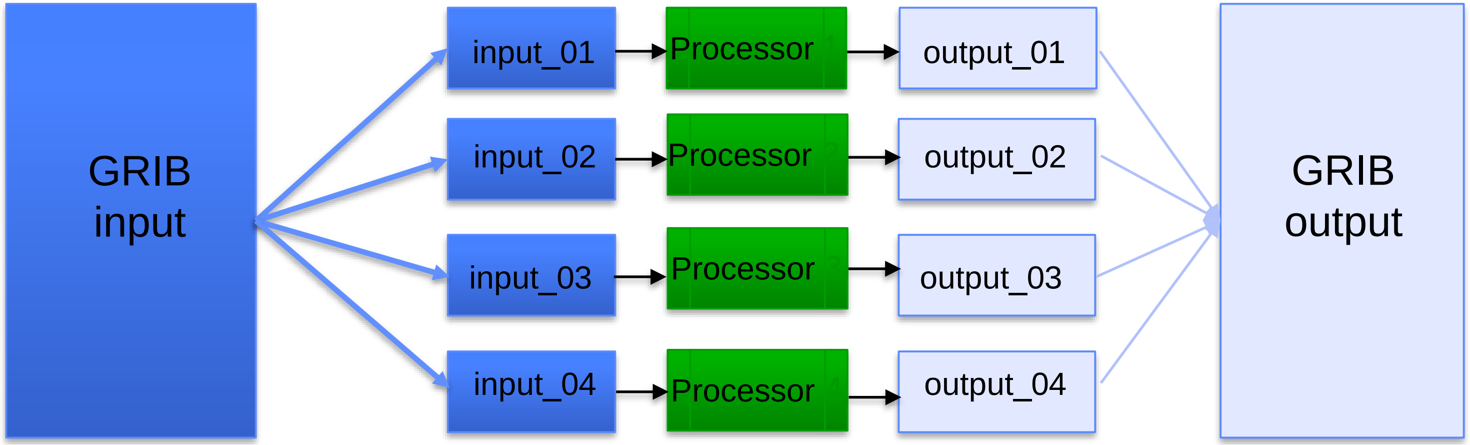
```
    codes_count file1 [file2 …]
```

# codes_split_file – splitting files and processing in parallel

- **Use codes_split_file to split an input GRIB file into chunks of roughly the same size**
- **The output files are called input_01, input_02, etc (where input is the name of the file)**
- **Much faster than grib_copy as no decoding of the header is done**
- **Syntax:**

```
codes_split_file [-v] nchunks input
```

- **Useful for parallelising operations where a large task is split into smaller ones which can be run on different processors**

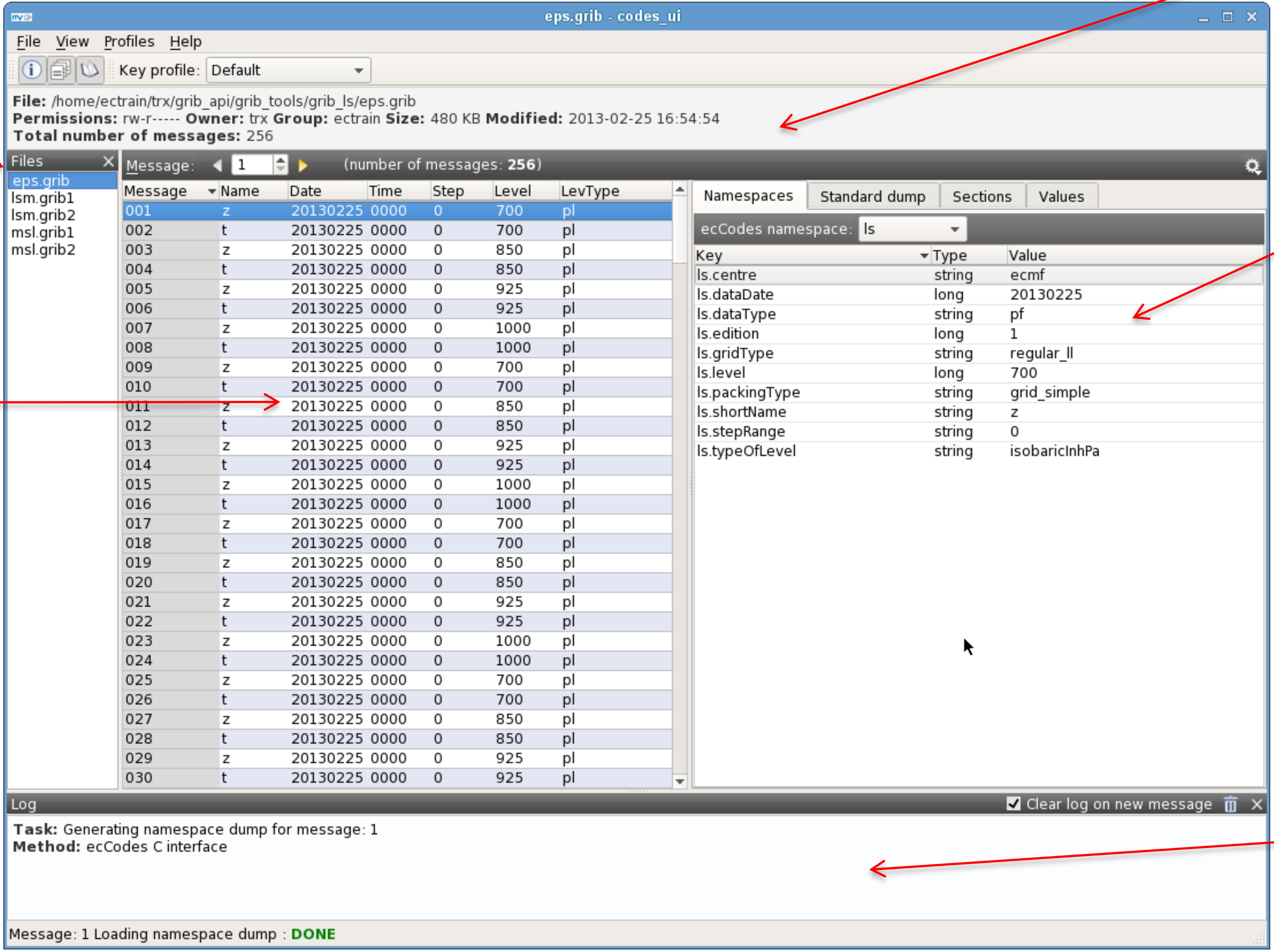


# CodesUI

- **CodesUI** is a standalone, UNIX-based graphical user interface built on ecCodes to handle GRIB (and BUFR) data to
  - Inspect the overall structure of GRIB files
  - Examine data and metadata of the individual messages
- CodesUI shares its codebase with the Metview code examiners. It was packaged as a standalone software application with the minimum possible dependencies requiring only ecCodes and Qt5 for installation.
- Can be started up from the command line. E.g. on ecgate use

```
codes_ui -g [grib_file_1 grib_file_2 …]
```

# CodesUI: The user interface



**File:** /home/ectrain/trx/grib_api/grib_tools/grib_ls/eps.grib
**Permissions:** rw-r----- **Owner:** trx **Group:** ectrain **Size:** 480 KB **Modified:** 2013-02-25 16:54:54
**Total number of messages:** 256

(number of messages: **256**)

| Message | Name | Date | Time | Step | Level | LevType |
|---|---|---|---|---|---|---|
| 001 | z | 20130225 | 0000 | 0 | 700 | pl |
| 002 | t | 20130225 | 0000 | 0 | 700 | pl |
| 003 | z | 20130225 | 0000 | 0 | 850 | pl |
| 004 | t | 20130225 | 0000 | 0 | 850 | pl |
| 005 | z | 20130225 | 0000 | 0 | 925 | pl |
| 006 | t | 20130225 | 0000 | 0 | 925 | pl |
| 007 | z | 20130225 | 0000 | 0 | 1000 | pl |
| 008 | t | 20130225 | 0000 | 0 | 1000 | pl |
| 009 | z | 20130225 | 0000 | 0 | 700 | pl |
| 010 | t | 20130225 | 0000 | 0 | 700 | pl |
| 011 | z | 20130225 | 0000 | 0 | 850 | pl |
| 012 | t | 20130225 | 0000 | 0 | 850 | pl |
| 013 | z | 20130225 | 0000 | 0 | 925 | pl |
| 014 | t | 20130225 | 0000 | 0 | 925 | pl |
| 015 | z | 20130225 | 0000 | 0 | 1000 | pl |
| 016 | t | 20130225 | 0000 | 0 | 1000 | pl |
| 017 | z | 20130225 | 0000 | 0 | 700 | pl |
| 018 | t | 20130225 | 0000 | 0 | 700 | pl |
| 019 | z | 20130225 | 0000 | 0 | 850 | pl |
| 020 | t | 20130225 | 0000 | 0 | 850 | pl |
| 021 | z | 20130225 | 0000 | 0 | 925 | pl |
| 022 | t | 20130225 | 0000 | 0 | 925 | pl |
| 023 | z | 20130225 | 0000 | 0 | 1000 | pl |
| 024 | t | 20130225 | 0000 | 0 | 1000 | pl |
| 025 | z | 20130225 | 0000 | 0 | 700 | pl |
| 026 | t | 20130225 | 0000 | 0 | 700 | pl |
| 027 | z | 20130225 | 0000 | 0 | 850 | pl |
| 028 | t | 20130225 | 0000 | 0 | 850 | pl |
| 029 | z | 20130225 | 0000 | 0 | 925 | pl |
| 030 | t | 20130225 | 0000 | 0 | 925 | pl |

ecCodes namespace: ls

| Key | Type | Value |
|---|---|---|
| ls.centre | string | ecmf |
| ls.dataDate | long | 20130225 |
| ls.dataType | string | pf |
| ls.edition | long | 1 |
| ls.gridType | string | regular_ll |
| ls.level | long | 700 |
| ls.packingType | string | grid_simple |
| ls.shortName | string | z |
| ls.stepRange | string | 0 |
| ls.typeOfLevel | string | isobaricInhPa |

**Task:** Generating namespace dump for message: 1
**Method:** ecCodes C interface

Message: 1 Loading namespace dump : **DONE**

File information

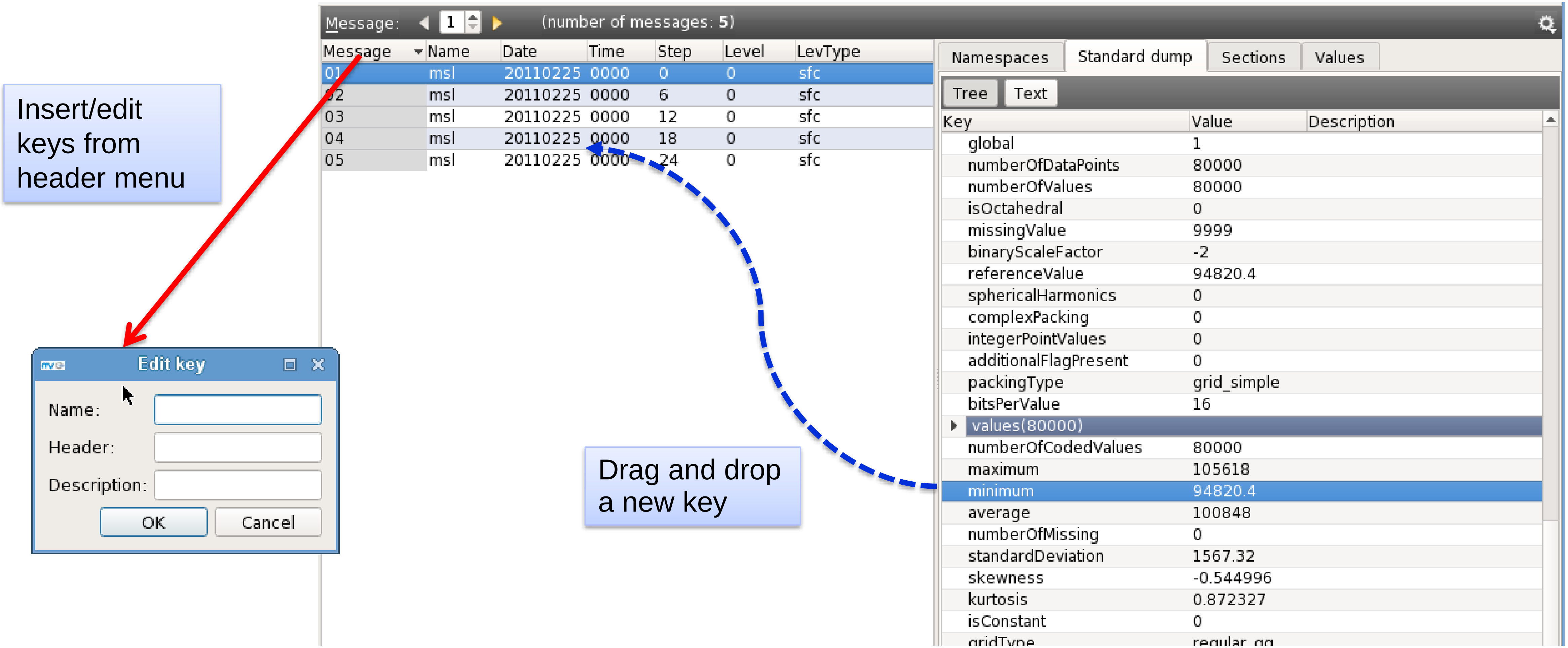
File list

Message content panel (grib_dump)

Message list (with user defined GRIB key selection)

Log

# CodesUI: Managing GRIB keys



Insert/edit keys from header menu

Drag and drop a new key

Message: 1 (number of messages: **5**)

| Message | Name | Date | Time | Step | Level | LevType |
|---|---|---|---|---|---|---|
| 01 | msl | 20110225 | 0000 | 0 | 0 | sfc |
| 02 | msl | 20110225 | 0000 | 6 | 0 | sfc |
| 03 | msl | 20110225 | 0000 | 12 | 0 | sfc |
| 04 | msl | 20110225 | 0000 | 18 | 0 | sfc |
| 05 | msl | 20110225 | 0000 | 24 | 0 | sfc |

| Key | Value | Description |
|---|---|---|
| global | 1 | |
| numberOfDataPoints | 80000 | |
| numberOfValues | 80000 | |
| isOctahedral | 0 | |
| missingValue | 9999 | |
| binaryScaleFactor | -2 | |
| referenceValue | 94820.4 | |
| sphericalHarmonics | 0 | |
| complexPacking | 0 | |
| integerPointValues | 0 | |
| additionalFlagPresent | 0 | |
| packingType | grid_simple | |
| bitsPerValue | 16 | |
| values(80000) | | |
| numberOfCodedValues | 80000 | |
| maximum | 105618 | |
| minimum | 94820.4 | |
| average | 100848 | |
| numberOfMissing | 0 | |
| standardDeviation | 1567.32 | |
| skewness | -0.544996 | |
| kurtosis | 0.872327 | |
| isConstant | 0 | |
| gridType | regular_gg | |

Edit key

Name:

Header:

Description:

# GRIB Tools Quiz

- **grib_get/grib_get_data**
- **grib_compare**

# Challenge 2: using grib_get, grib_get_data and grib_compare (1)

1. Use grib_get to print the shortName, dataTime, dataDate and level for the 500 &1000 hPa levels only **in tz_an_pl.grib1**
2. Use grib_get to print the stepRange for the fields in the file surface.grib1 in (a) hours (b) minutes and (c) seconds – what happens ?
3. Use grib_get_data to print the latitude, longitude and values for the first (–w count=1**) field** in surface.grib1
   - Output the data values in decimal format with 5 decimal places
   - Output the data values in exponential format with 10 decimal places
   - **Are there any missing values ?**
4. Use grib_get_data to print the data values for the temperature at 500 hPa only from the file tz_an_pl.grib1
   - Make sure you print only the data for T at 500 hPa ! What is printed ?

# Challenge 2: using grib_get, grib_get_data and grib_compare (2)

5. Use grib_compare to compare the headers of the GRIB messages contained in the files file1.grib and file2.grib
   - **Use the "-H" option to restrict the comparison to the headers only**
   - Which keys does grib_compare report as different ?
   - What is the exit code returned ?
6. Compare the data namespaces (use "-c data:n") for file1.grib and file2.grib.
   - What values need to be set for the absolute (with `-A`) and relative (with `-R`) error tolerances for the comparison to be successful ?
   - **How many data values compare to within twice the packing error ?**

# GRIB Tools Quiz

- **grib_copy**
- **grib_set**
- **grib_to_netcdf**

# Challenge 3: modifying GRIB messages

1. The file **file1**.grib1 contains parameters T and Z on six pressure levels.

   - Use grib_copy to create two files, one containing all the pressure levels for parameter T, the other for Z. Check the content of the new files with grib_ls

   - **Repeat but output the messages so the levels in the new files are in increasing numerical order**

2. Use grib_set to change the date and time to 12UTC on 04 February 2019 for all messages in file1.grib1

   - **Repeat but change the date and time for T at 500 hPa only**

   - **Repeat so that T at 500 hPa only is written to the output file**

3. Use grib_to_netcdf to convert the GRIB messages in file2.grib1 to NetCDF.

   - Try with both the default data type (NC_SHORT) and NC_FLOAT. Check the data values in each case with ncdump.

   - **Repeat but set the Reference date to 6 February 2019 and compare the time variable with previous results**

4. Use grib_to_netcdf to convert the GRIB messages in file3.grib1 to NetCDF.

   - What happens … and why ?

# GRIB Tools Quiz

- **grib_filter**

# Challenge 4 – using grib_filter

1. Run grib_filter with the rules files ‘print.filter’, ‘write.filter’, ‘transient.filter’ on ‘tigge.grib’.
2. Comment/uncomment the instructions one by one to see the different behaviours.

**Advanced**

3. Change the date to 20170301 and the step to step+48 in the file ‘tigge.grib’ only for the data produced by ECMWF. Write all messages to a file called **‘question1.grib’.**
4. Set the values of the first message (remember the count key !) in the file ‘tigge.grib’ to 1.2, 3.4, 5.6, 3.7 and step to 72. Write only this message to the file ‘**question2.grib**’. Check the values coded with grib_get_data or grib_dump.
5. Append to ‘**question2.grib**’ all the messages containing the same parameter of the other centres that are not encoded using a reduced Gaussian grid, setting the step to 72.

You can check if your filter is correct by comparing your output GRIB file with the sample in sample_outputs/, e.g.:

```
> grib_filter -o question1.grib question1.filter tigge.grib
> grib_compare question1.grib sample_outputs/question1.grib
```

# Advanced grib_filter challenge

6. Split ‘tigge.grib’ into several files, one for each centre, containing only surface parameters and parameters that are at level 10 of height above ground.

   - For the surface parameters, set changeDecimalPrecision to 2,
   - **F**or the height above ground parameters set **changeDecimalPrecision** to 3.

   Print information messages for each case, such as:

```
Centre ammc parameter v not written
Centre ammc parameter 10u written to question4-ammc.out
```

7. Merge the messages from the previously split GRIB files into a single file

   - Write only messages encoded in a regular lat-long grid, and exclude messages where the parameters are 10u or 10v.