

# Using GRIB Tools

## Computer User Training Course 2018

**Paul Dando & Carsten Maass**

User Support

[advisory@ecmwf.int](mailto:advisory@ecmwf.int)



# Contents

- **GRIB Tools basics and getting help**
- **Information tools**
- **Inspection tools**
- **Getting key / value pairs**
- **Getting data values**
- **Comparing messages**
- **Modification tools**
- **Copying messages**
- **Setting key / value pairs**
- **Converting GRIB to netCDF**

## ecCodes command line tools – basic concepts

- The ecCodes tools are a set of command line programs for interactive and batch processing of GRIB data
- Provide ready and tested solutions to the most common processing of GRIB data
- Their use will avoid the need to write new code and thus speed up your work
  - Consider using ecCodes tools instead of writing your own program
- The tools are provided with a common set of options so that it is quick to apply the same options to different tools
  
- Use of the tools is recommended whenever possible!

# GRIB Tools – basics

All tools use a common syntax

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

There are tools to

- count the messages in a GRIB file
  - `grib_count`
- inspect the content of and compare GRIB files
  - `grib_ls`, `grib_dump`, `grib_get`, `grib_get_data`, `grib_compare`
- copy some messages
  - `grib_copy`
- change the content of a GRIB message
  - `grib_set`, `grib_filter`
- convert a GRIB file to netCDF format
  - `grib_to_netcdf`

## Getting help

- **UNIX 'man'-style pages are available for each tool by running the tool without any options or input file**

```
> grib_dump
```

```
NAME      grib_dump
```

```
DESCRIPTION
```

```
    Dump the content of a grib file in different formats.
```

```
USAGE
```

```
grib_dump [options] grib_file grib_file ...
```

```
OPTIONS
```

```
-O      Octet mode. WMO documentation style dump.  
-D      Debug mode.  
-d      Print all data values.
```

```
...
```

# Generic ecCodes tools

- There is a tool for getting information about the ecCodes installation
  - [codes\\_info](#)
- There is a tool for counting GRIB and BUFR messages
  - [codes\\_count](#)

## 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 2.6.0
```

```
Default definition files path is used:
```

```
/usr/local/apps/eccodes/2.6.0/GNU/5.3.0/share/eccodes/definitions
```

```
Definition files path can be changed setting ECCODES_DEFINITION_PATH environment variable
```

```
Default SAMPLES path is used:
```

```
/usr/local/apps/eccodes/2.6.0/GNU/5.3.0/share/eccodes/samples
```

```
SAMPLES path can be changed setting ECCODES_SAMPLES_PATH environment variable
```

# ecCodes documentation

- The ecCodes documentation and support pages are available at <https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>
- The GRIB Tools are documented at <https://software.ecmwf.int/wiki/display/ECC/GRIB+tools>  
Includes some examples of how to use the tools
- The ecCodes software can be downloaded from <https://software.ecmwf.int/wiki/display/ECC/Releases>



# ecCodes keys and parameters for GRIB – THE Reference

- Parameters in GRIB

- GRIB Parameter Database - <http://apps.ecmwf.int/codes/grib/param-db>

- ecCodes GRIB keys - <http://apps.ecmwf.int/codes/grib/>

- GRIB Edition 1 - <http://apps.ecmwf.int/codes/grib/format/grib1/>

- GRIB Edition 2 - <http://apps.ecmwf.int/codes/grib/format/grib2/>

- GRIB Edition Independent - <http://apps.ecmwf.int/codes/grib/format/edition-independent/>

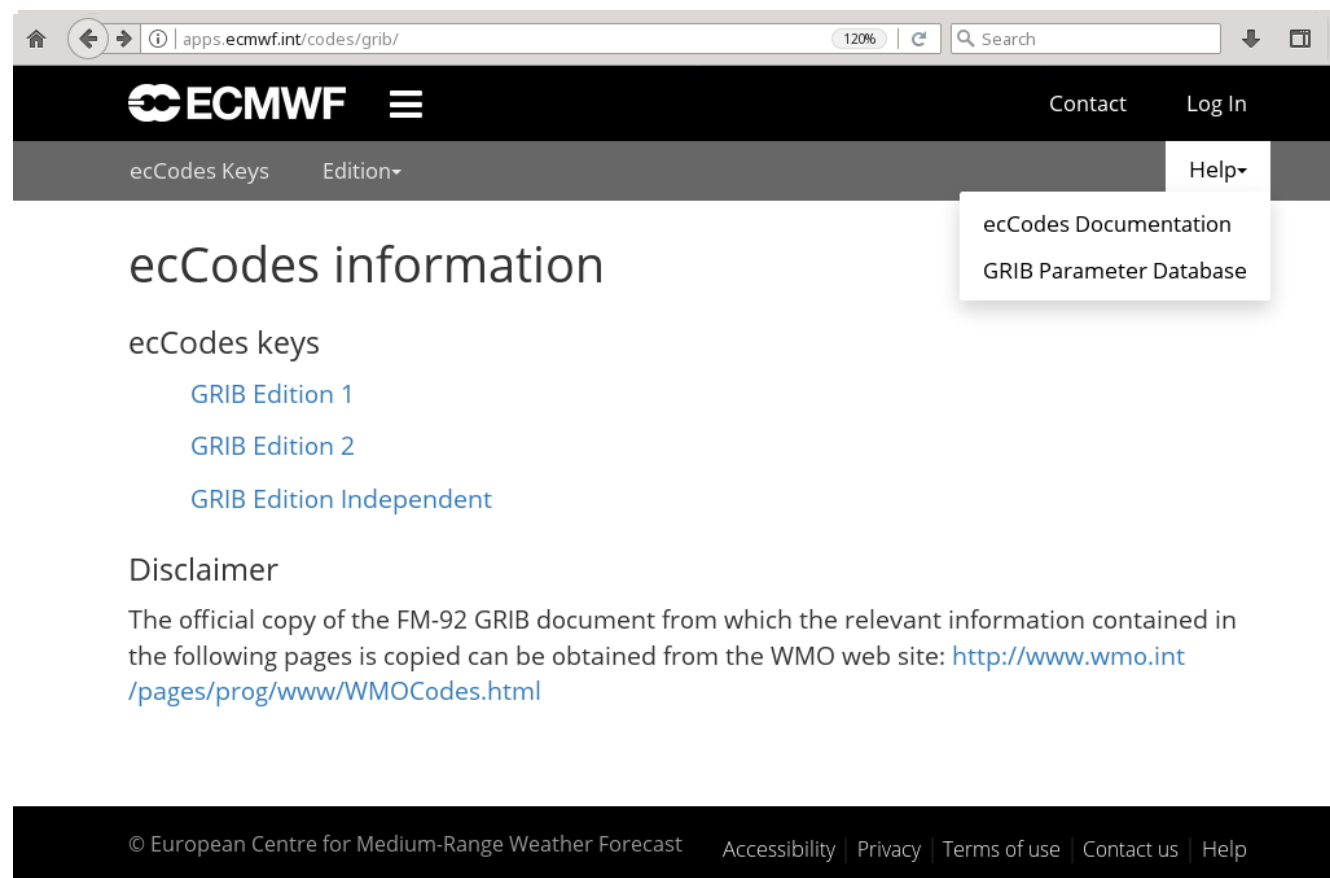
## Disclaimer

*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 keys

- For definitions of edition independent keys, GRIB1 or GRIB2 keys see <http://apps.ecmwf.int/codes/grib/>
- Usage of edition independent keys should be preferred



The screenshot shows a web browser window at the URL [apps.ecmwf.int/codes/grib/](http://apps.ecmwf.int/codes/grib/). The page features the ECMWF logo and navigation links for 'Contact' and 'Log In'. A 'Help' dropdown menu is open, showing 'ecCodes Documentation' and 'GRIB Parameter Database'. The main content area is titled 'ecCodes information' and lists 'ecCodes keys' with links for 'GRIB Edition 1', 'GRIB Edition 2', and 'GRIB Edition Independent'. A 'Disclaimer' section follows, stating that the official copy of the FM-92 GRIB document is available at <http://www.wmo.int/pages/prog/www/WMOCodes.html>. The footer contains copyright information for the European Centre for Medium-Range Weather Forecast and links for 'Accessibility', 'Privacy', 'Terms of use', 'Contact us', and 'Help'.

# The parameter database

- The parameter database stores information about the GRIB 1, GRIB 2 encoding of all parameters recognised by ecCodes

The screenshot shows the ECMWF GRIB Parameter Database web interface. The header includes the ECMWF logo and navigation links for 'Contact', 'Log In', and 'Help'. The main content area is titled 'Parameter database:' and features a search bar with the text 'temp' and a 'Show 20 entries' dropdown. Below the search bar is a table of parameters with columns for Name, Short Name, Units, Parameter ID, GRIB1, and GRIB2. The table lists several temperature-related parameters, all with units in Kelvin (K) and GRIB1/GRIB2 status marked with checkmarks. To the left of the table are filter options for 'Filter' (All, GRIB Edition 1, GRIB Edition 2), 'Discipline' (All), and 'Category' (All).

Name	Short Name	Units	Parameter ID	GRIB1	GRIB2
Potential temperature	pt	K	3	✓	✓
Equivalent potential temperature	eqpt	K	4	✓	✓
Saturated equivalent potential temperature	sept	K	5	✓	✓
Unbalanced component of temperature	uctp	K	21	✓	✓
Sea surface temperature	sst	K	34	✓	✓
Ice temperature layer 1	istl1	K	35	✓	✓

- The database is accessible via a web interface at:  
<http://apps.ecmwf.int/codes/grib/param-db>

## `grib_count` – count GRIB messages

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

```
grib_count grib_file1 [grib_file2 ...]
```

**(takes wildcards)**

## `grib_dump` – dump content of GRIB files

- Use **`grib_dump`** to dump the content of a file containing one or more GRIB messages
- Various output formats are supported
  - **Octet mode** provides a WMO documentation style dump
  - **Debug mode** prints all keys available in the GRIB file
  - **Octet** and **Debug modes** cannot be used together
  - Octet content can also be printed in hexadecimal format
- Options also exist to print key **aliases** and key **type** information
- Output to JSON (JavaScript Object Notation)
  - Easy to process

# grib\_dump – usage

```
grib_dump [options] grib_file grib_file ...
```

## Basic options

-O	Octet mode (WMO documentation style)
-D	Debug mode
-a	Print key alias information
-t	Print key type information
-H	Octet content in Hexadecimal
-D	Debug mode
-w key[:{s i d}] {= !=}value,...	Where clause
-j	JSON output
-V	Print ecCodes Version

## grib\_dump – examples

```
> grib_dump file.grib1
```

```
***** FILE: file.grib1
#===== MESSAGE 1 ( length=3280398 ) =====
GRIB {
  editionNumber = 1;
  table2Version = 128;
  # European Center for Medium-Range Weather Forecasts (grib1/0.table)
  centre = 98;
  generatingProcessIdentifier = 139;
  # Geopotential (m**2 s**-2) (grib1/2.98.128.table)
  indicatorOfParameter = 129;
  # Isobaric level pressure in hectoPascals (hPa) (grib1/3.table)
  indicatorOfTypeOfLevel = 100;
  level = 1000;
  # Forecast product valid at reference time + P1 (P1>0) (grib1/5.table)
  timeRangeIndicator = 0;
  # Unknown code table entry (grib1/0.ecmf.table)
  subCentre = 0;
  paramId = 129;
  #-READ ONLY- units = m**2 s**-2;
  #-READ ONLY- nameECMF = Geopotential;
  #-READ ONLY- name = Geopotential;
  decimalScaleFactor = 0;
  dataDate = 20110223;
  dataTime = 1200; ...
```

## `grib_dump` – examples

```
> grib_dump -O file.grib1
```

```
***** FILE: file.grib1
```

```
#===== MESSAGE 1 ( length=3280398 )
```

```
=====
```

```
1-4 identifier = GRIB
```

```
5-7 totalLength = 3280398
```

```
8 editionNumber = 1
```

```
===== SECTION_1 ( length=52, padding=0
```

```
=====
```

```
1-3 section1Length = 52
```

```
4 table2Version = 128
```

```
5 centre = 98 [European Centre for Medium-Range Weather Forecasts (grib1/0.table) ]
```

```
6 generatingProcessIdentifier = 145
```

```
7 gridDefinition = 255
```

```
8 section1Flags = 128 [10000000]
```

```
9 indicatorOfParameter = 129 [Geopotential (m**2 s**-2) (grib1/2.98.128.table) ]
```

```
10 indicatorOfTypeOfLevel = 100 [Isobaric level pressure in hectoPascals(hPa)
```

```
(grib1/local/ecmf/3.table , grib1/3.table) ]
```

```
11-12 level = 1000
```

```
13 yearOfCentury = 16
```

```
14 month = 2
```

```
15 day = 27
```

```
16 hour = 12
```

```
17 minute = 0
```

```
18 unitOfTimeRange = 1 [Hour (grib1/4.table) ]
```

```
19 P1 = 0 ...
```



# grib\_dump – examples

```
> grib_dump -OtaH file.grib1
```

```
***** FILE: file.grib1
#===== MESSAGE 1 ( length=3280398 ) =====
1-4      ascii identifier = GRIB ( 0x47 0x52 0x49 0x42 )
5-7      g1_message_length totalLength = 3280398 ( 0x32 0x0E 0x0E )
8        unsigned editionNumber = 1 ( 0x01 ) [ls.edition]
===== SECTION_1 ( length=52, padding=0 ) =====
1-3      section_length sectionLength = 52 ( 0x00 0x00 0x34 )
4        unsigned table2Version = 128 ( 0x80 ) [gribTablesVersionNo]
5        codetable centre = 98 ( 0x62 ) [European Center for Medium-Range Weather
        Forecasts (grib1/0.table) ] [identificationOfOriginatingGeneratingCentre, originatingCentre,
        ls.centre, centreForTable2]
6        unsigned generatingProcessIdentifier = 139 ( 0x8B ) [generatingProcessIdentificationNumber, process]
7        unsigned gridDefinition = 255 ( 0xFF )
8        codeflag section1Flags = 128 [10000000] ( 0x80 )
9        codetable indicatorOfParameter = 129 ( 0x81 ) [Geopotential (m**2 s**-2) (grib1/2.98.128.table) ]
10       codetable indicatorOfTypeOfLevel = 100 ( 0x64 ) [Isobaric level pressure in hectoPascals (hPa)
        (grib1/3.table) ] [levelType, mars.levtype]
11-12    unsigned level = 1000 ( 0x03 0xE8 ) [vertical.topLevel, vertical.bottomLevel, ls.level, lev,
        mars.levelist]
13       unsigned yearOfCentury = 11 ( 0x0B )
14       unsigned month = 2 ( 0x02 )
15       unsigned day = 23 ( 0x17 )
16       unsigned hour = 12 ( 0x0C )
17       unsigned minute = 0 ( 0x00 ) . . .
```

# grib\_dump – examples

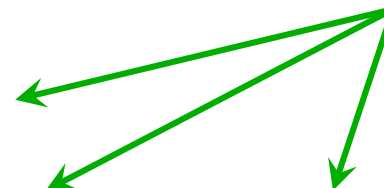
```
> grib_dump -D file.grib1
```

```
***** FILE: file.grib1
#===== MESSAGE 1 ( length=9358 )
:
=====> section GRIB (9358,9358,0)
0-0 constant ieeeFloats = 0
=====> section section_0 (0,0,0)
----> label empty
<==== section section_0
0-4 ascii identifier = GRIB
4-7 gl_message_length totalLength = 9358
7-8 unsigned editionNumber = 1 [ls.edition]
=====> section section_1 (52,52,0)
:
36-36 gldate dataDate = 20110223 [mars.date, time.dataDate]
36-36 evaluate year = 2011
36-36 time dataTime = 1200 [mars.time]
36-36 julian_day julianDay = 2.45562e+06
36-36 codetable stepUnits = 1 [Hour (stepUnits.table) ]
36-36 concept stepType = instant
36-36 glstep_range stepRange = 0 [time.stepRange]
36-36 long_vector startStep = 0
36-36 long_vector endStep = 0 [stepInHours, mars.step]
36-36 mars_param marsParam = 129.128 [mars.param]
36-36 validity_date validityDate = 20110223
36-36 validity_time validityTime = 1200
...

```

*In debug mode computed keys are shown*

*ls.<key>, mars.<key> and time.<key> denote keys in namespaces*



## ecCodes – namespaces

namespace	keys
ls	centre, shortName, level etc. used by the grib_ls tool
parameter	paramId, shortName, units etc. which relate to the meteorological parameter
statistics	maximum, minimum, average, standard deviation etc. related to the statistics of the data values
time	forecast date, validity date, steps etc. describing the forecast runs
geography	bounding box of the grid, number of points along a parallel etc. describing the grid geometry
vertical	type of the level, list of coefficients of the vertical coordinate etc. describing the levels and layers
mars	class, stream, type etc. used to describe the content of MARS (ECMWF's Meteorological Archival and Retrieval System)

Particularly useful with `grib_ls`

# Practical

- Work in your \$SCRATCH

```
cd $SCRATCH
```

- Make a copy of the practicals 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 practicals

- There is a sub-directory for each practical:

```
ls $SCRATCH/grib_tools
```

```
grib_compare  grib_copy    grib_dump    grib_get    grib_ls
```

```
grib_set     . . .
```

# Practical: using grib\_dump

- Familiarise yourself with the web documentation by exploring the different keys available for type GRIB1 and type GRIB2 messages
  - Identify some keys common to both GRIB1 and GRIB2
- 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?

## `grib_ls` – list the content of GRIB files

- Use `grib_ls` to list the content of GRIB files
- Without options `grib_ls` prints a default list of keys
  - The default list printed is different for GRIB 1 and GRIB 2
- Options exist to specify the set of keys to print or to print keys in addition to the default set
- Output can be ordered
  - e.g. order by ascending or descending step
- `grib_ls` does not fail if a key is not found
- `grib_ls` can also be used to find the grid point(s) nearest to a specified latitude-longitude and print the value of the field at that point(s)
  - Modes available to obtain one or four nearest grid points

## `grib_ls` – usage

```
grib_ls [options] grib_file grib file ...
```

### Options

- `-p key[:{s|i|d}],...` Keys to print
- `-P key[:{s|i|d}],...` Additional keys to print
- `-w key[:{s|i|d}]{=|!=}value,...` Where clause
- `-B "key asc, key desc..."` Order by: "step asc, centre desc"
- `-n namespace` Print all the keys belonging to namespace  
(ls, parameter, statistics, geography, time, mars, vertical)
- `-i index` Data value corresponding to given index is printed
- `-m` Print MARS keys (short for `-n mars`)
- `-W width` Minimum column width (default 10)
- ...

# grib\_ls – examples

Use **-p** option to specify a list of keys to be printed:

```
> grib_ls file.grib2
file.grib2
edition centre date      ... gridType  ...  typeOfLevel  level  shortName  packingType
2         ecmf   20110226 ... reduced_gg ...  isobaricInhPa 1000   q          grid_simple
2         ecmf   20110226 ... reduced_gg ...  isobaricInhPa 850    q          grid_simple
2         ecmf   20110226 ... reduced_gg ...  isobaricInhPa 700    q          grid_simple
2         ecmf   20110226 ... reduced_gg ...  isobaricInhPa 500    q          grid_simple
4 of 4 grib messages in file1.grib2

4 of 4 total grib messages in 1 files
```

```
> grib_ls -p centre:i,dataDate,shortName,paramId,typeOfLevel,level file.grib2
file.grib2
Centre      dataDate      shortName      paramId      typeOfLevel      level
98          20110226      q              133          isobaricInhPa    1000
98          20110226      q              133          isobaricInhPa    850
98          20110226      q              133          isobaricInhPa    700
98          20110226      q              133          isobaricInhPa    500
4 of 4 grib messages in file.grib1

4 of 4 total grib messages in 1 files
```



## grib\_ls – examples

- When a key is not present in the GRIB file, it returns “not found” for this key

```
> grib_ls -p my_key file.grib1

file.grib1
my_key
not found

> echo $?
0
```

*exit code returned = 0*

- Similar behaviour to grib\_get (see later)
  - grib\_ls is better for interactive use
  - use grib\_get within scripts

## Using the 'where' option

- The 'where option' `-w` can be used with all GRIB Tools

- Constraints are of the form `key=value` or `key!=value`

```
-w key[:{s|i|d}]=value, key[:{s|i|d}]!=value
```

- Messages are processed only if they match ALL key/value constraints
- Values separated by / represent "OR" condition

```
> grib_ls -w levelType=pl file.grib1
...
> grib_ls -w step!=6,level=700/850 file.grib1
...
> grib_ls -w count=3 file.grib1
```

# Practical: using grib\_ls

- Use **grib\_ls** to inspect the files **msl.grib1** and **msl.grib2**
  - Which keys does **grib\_ls** show by default for the two files ?
  - What fields do they contain ?
- Use **grib\_ls** to print the MARS keys
- Use **grib\_ls** with other namespaces
- Use **grib\_ls** to order the output by descending step
- Use **grib\_ls** to print the **centre**, **dataDate**, **stepRange**, **levelType**, **shortName** and **paramId** for both files
  - Experiment with both **-P** and **-p** options and **'key:i'**, **'key:s'**

## Finding nearest grid points with grib\_ls

- The value(s) of a GRIB field close to the point of a Latitude/Longitude can be found with **grib\_ls**

```
grib_ls -l Latitude,Longitude[,MODE,file] grib_file
```

**MODE** Can take the values

- 4** Print values at the 4 nearest grid points (default)
- 1** Print value at the closest grid point

**file** Specifies a GRIB file to use as a mask  
The closest *land* point (with mask  $\geq 0.5$ ) is printed

- GRIB files specified **must** contain grid point data

## Practical: using grib\_ls -l

- The file `msl.grib1` contains the mean sea-level pressure from the ENS control forecast at 6-hourly time steps for the first 24 hours on a N100 regular Gaussian grid
- 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 ?
- Change the command used to output only the forecast step and the MSLP value at the nearest grid point
- Change the command to output the MSLP values at the four grid points nearest to ECMWF
- Use the file `lsm.grib1` to provide a land-sea mask
  - Are all four nearest grid points land points (mask  $\geq 0.5$ ) ?

# 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 is still a beta version!*

# CodesUI: The user interface

The screenshot shows the CodesUI application window titled "eps.grib - codes\_ui". The interface includes a menu bar (File, View, Profiles, Help), a toolbar, and a main workspace. A callout box labeled "File list" points to the left sidebar showing a list of files: eps.grib, lsm.grib1, lsm.grib2, msl.grib1, and msl.grib2. A callout box labeled "File information" points to the top status bar showing file details: 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. A callout box labeled "Message list (with user defined GRIB key selection)" points to the central table of messages. A callout box labeled "Message content panel (grib\_dump)" points to the right-hand panel displaying a table of key-value pairs for the selected message. A callout box labeled "Log" points to the bottom status bar showing the task: "Task: Generating namespace dump for message: 1" and "Method: ecCodes C interface".

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

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

# CodsUI: Managing GRIB keys

Insert/edit keys from header menu

**Edit key**

Name:

Header:

Description:

OK Cancel

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

Namespaces Standard dump Sections Values

Tree Text

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 ga	

Drag and drop a new key