grib_set

DESCRIPTION

Sets key/value pairs in the input GRIB file and writes each message to the output_grib_file. It fails when an error occurs (e.g. key not found).

USAGE

grib_set [options] grib_file grib_file ... output_grib_file

OPTIONS

-s key[:{s|d|i}]=value,key[:{s|d|i}]=value,...

Key/values to set. For each key a string (key:s), a double (key:d) or an integer (key:i) type can be defined. By default the native type is set.

-r

Repack data. Sometimes after setting some keys involving properties of the packing algorithm a repacking of data is needed. This repacking is performed setting this -r option.

-d value

Set all the data values to "value".

-p key[:{s|d|i}],key[:{s|d|i}],...

Declaration of keys to print. For each key a string (key:s), a double (key:d) or an integer (key:i) type can be requested. Default type is string.

-P key[:{s|d|i}],key[:{s|d|i}],...

As -p adding the declared keys to the default list.

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

Where clause. Set is only executed for GRIB messages matching all the key/value constraints. If a GRIB message does not match the constraints it is copied unchanged to the output_grib_file. This behaviour can be changed setting the option -S. A valid constraint is of type key=value or key!=value. For each key a string (key:s), a double (key:d) or an integer (key:i) type can be defined. Default type is string. In the value you can also use the forward-slash character '/' to specify an OR condition (i.e. a logical disjunction) Note: only one -w clause is allowed.

-7

Does not fail when the message has wrong length

-S

Strict. Only messages matching all the constraints are copied to the output file

-V

Version.

-M

Multi-field support off. Turn off support for multiple fields in single GRIB message.

-g

Copy GTS header.

-G

GRIBEX compatibility mode.

-T T | B | M | A

Message type. T->GTS, B->BUFR, M->METAR (Experimental),A->Any (Experimental). The input file is interpreted according to the message type.

-f

Force. Force the execution not to fail on error.

-v

Verbose.

grib_set examples

1. To set productDefinitionTemplateNumber=2 only for the fields with productDefinitionTemplateNumber=11

```
> grib_set -s productDefinitionTemplateNumber=2 -w
productDefinitionTemplateNumber=11 ../data/tigge_pf_ecmwf.grib2
out.grib2
```

2. To set productDefinitionTemplateNumber=2 only for the fields for which productDefinitionTemplateNumber is not equal to 11

```
> grib_set -s productDefinitionTemplateNumber=2 -w
productDefinitionTemplateNumber!=11 tigge_pf_ecmwf.grib2 out.grib2
```

3. When a key is not used all the bits of its value should be set to 1 to indicate that it is missing. Since the length (number of octet) is different from a key to another, the value that we have to code for missing keys is not unique. To give an easy way to set a key to missing a string "missing" or "MISSING" is accepted by grib_set as follows:

```
> grib_set -s scaleFactorOfFirstFixedSurface=missing,
scaledValueOfFirstFixedSurface=MISSING ../data
/regular_latlon_surface.grib2 out.grib2
```

Since some values can not be set to missing you can get an error for those keys.

4. To set scaleFactorOfSecondFixedSurface to missing only for the fields for which scaleFactorOfSecondFixedSurface is not missing:

```
> grib_set -s scaleFactorOfSecondFixedSurface=missing -w
scaleFactorOfSecondFixedSurface!=missing tigge_pf_ecmwf.grib2 out.
grib2
```

5. It is possible to produce a GRIB edition 2 file from a GRIB edition 1 by just changing the edition number with grib_set. However it is important that you carefully inspect the output and check the information is correctly translated.

```
grib_set -s edition=2 ../data/reduced_gaussian_pressure_level.
grib1 out.grib2
```

For more details please see: https://software.ecmwf.int/wiki/display/ECC/GRIB%3A+Converting+edition+1+to+2

6. With grib edition 2 is possible to compress data using the jpeg algorithm. To change packing algorithm from grid_simple (simple packing) to grid_jpeg (jpeg2000 packing):

```
> grib_set -s packingType=grid_jpeg ../data
/regular_gaussian_model_level.grib2 out.grib2
```

7. It's possible to ask ecCodes to calculate the number of bits per value needed to pack a given field with a fixed number of decimal digits of precision. For example if we want to pack a temperature expressed in Kelvin with 1 digits of precision after the decimal point we can set changeDecimalPrecision=1

```
> grib_set -s changeDecimalPrecision=1 ../data
/regular_latlon_surface.grib2 ../data/out.grib2
```