

# **grib\_compare**

## **DESCRIPTION**

Compare GRIB messages contained in two files. If some differences are found it fails returning an error code. Floating-point values are compared exactly by default, different tolerance can be defined see -P -A -R. Default behaviour: absolute error=0, bit-by-bit compare, same order in files.

## **USAGE**

`grib_compare [options] grib_file1 grib_file2`

## **OPTIONS**

**-r**

Compare files in which the messages are not in the same order. This option is time expensive.

**-b key,key,...**

All the keys in this list are skipped in the comparison. Bit-by-bit compare on.

**-e**

Edition independent compare. It is used to compare GRIB edition 1 and 2.

**-2**

Enable two-way comparison.

**-c key[:i|d|s|n],key[:i|d|s|n],...**

Only the listed keys or namespaces (:n) are compared. The optional letter after the colon is used to force the type in the comparison: i->integer, d->float, s->string, n->namespace. See -a option. Incompatible with -H option.

**-S start**

First field to be processed.

**-E end**

Last field to be processed.

**-a**

-c option modifier. The keys listed with the option -c will be added to the list of keys compared without -c.

**-H**

Compare only message headers (everything except data and bitmap). Bit-by-bit compare on. Incompatible with -c option.

**-R key1=relative\_error1,key2=relative\_error2,...**

Compare floating-point values using the relative error as tolerance. key1=relative\_error1 will compare key1 using relative\_error1. all=relative\_error will compare all the floating-point keys using relative\_error. Default all=0.

**-A absolute error**

Compare floating-point values using the absolute error as tolerance. Default is absolute error=0

**-P**

Compare data values using the packing error as tolerance.

**-T factor**

Compare data values using factor multiplied by the tolerance specified in options -P -R -A.

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

Where clause. Messages are processed only if they match all the key/value constraints. 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 specified. 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.

**-f**

Force. Force the execution not to fail on error.

**-V**

Version.

**-7**

Does not fail when the message has wrong length

**-v**

Verbose.

## **grib\_compare examples**

1. The default behaviour for `grib_compare` without any option is to perform a bit by bit comparison of the two messages. If the messages are found to be bitwise different then `grib_compare` switches to a "key based" mode to find out which coded keys are different. To see how `grib_compare` works we first set the `shortName=2d` (2 metre dew point temperature) in the file `regular_latlon_surface.grib1`

```
> grib_set -s shortName=2d regular_latlon_surface.grib1 2d.grib1
```