

How-to Articles

- [GRIB Howtos](#)
- [BUFR Howtos](#)

GRIB Howtos

- [How can I convert a multi-field GRIB file to single-field - ecCodes GRIB FAQ](#)
- [Why do I get a bitmap even though none of my values are missing - ecCodes GRIB FAQ](#)
- [How do I use an OR condition \(logical disjunction\) in a where clause - ecCodes GRIB FAQ](#) — For example in `grib_get` you may want to show all messages which have a level of 100, 150 or 200.
- [Memory-based access to definition/sample files \(MEMFS\) - ecCodes FAQ](#) — The definition and sample files used by ecCodes engine to decode all kind of messages can be built into the library (embedded) providing a performance improvement by significantly reducing access to the filesystem
- [Creating your own local definitions - ecCodes GRIB FAQ](#) — Local definitions are required when one needs to store information which is not part of the standard GRIB header. This might be local parameters or local definitions.
- [How can I find out which keys can be set to "missing" - ecCodes GRIB FAQ](#)
- [How do I set the product definition template in a GRIB2 message - ecCodes GRIB FAQ](#) — Let's say you want your message to use the product definition template 40 (The full list is defined in Code table 4.0):
- [How do I know if I am dealing with a multi-field GRIB file - ecCodes GRIB FAQ](#) — The only possible way of detecting if a GRIB file is multi-field is by using the `grib` tools or the API to turn the support for multiple fields in a single GRIB message on or off and observe the results
- [How can I multiply the field values by a constant - ecCodes GRIB FAQ](#) — One way is to use `grib_set` with the key "scaleValuesBy". Let's say you have a field in `orig.grib` whose values you want to multiply by 2.1. First check the original data value statistics
- [How can I remove the PV array \(list of vertical coordinates\) - ecCodes GRIB FAQ](#) — This can be done by setting the key "deletePV" as shown:

```
% grib_set -s deletePV=1 in.grib out.grib
```
- [grib_compare: index based comparison - ecCodes GRIB FAQ](#) — Suppose you have two large GRIB files and their messages are not in order, how do you compare them? One technique is to build index files from inputs and then compare those index files.
- [What are Code and Flag tables - ecCodes GRIB and BUFR FAQ](#)
- [How can I swap the scanning order - ecCodes GRIB FAQ](#)
- [How can I see the contents of an index file - ecCodes GRIB FAQ](#) — You can use the `grib_dump` command with the "-D" option:
- [Order of setting fixed surface keys - ecCodes GRIB FAQ](#)

BUFR Howtos

- [Memory-based access to definition/sample files \(MEMFS\) - ecCodes FAQ](#) — The definition and sample files used by ecCodes engine to decode all kind of messages can be built into the library (embedded) providing a performance improvement by significantly reducing access to the filesystem
- [What are Code and Flag tables - ecCodes GRIB and BUFR FAQ](#)
- [How can I encode pressure with higher precision - ecCodes BUFR FAQ](#)
- [Dump instructions to decode the input - ecCodes BUFR FAQ](#) — There is a "-D" option in the `bufr_dump` tool to generate instructions to read (**decode**) the input BUFR message.
- [Dump instructions to create the input - ecCodes BUFR FAQ](#) — There is a "-E" option in the `bufr_dump` tool to generate instructions to create (**encode**) the input BUFR message.
- [Performance improvement by skipping some keys - ecCodes BUFR FAQ](#) — When we set the "unpack" key to decode the data section, for every Table B element key we create a set of attributes. So for example examine the output from "`bufr_dump -jf`" and you see meta-data attributes like
- [How do I create BUFR from a CSV - ecCodes BUFR FAQ](#)
- [Adding your own local definition - ecCodes BUFR FAQ](#)
- [Add/remove local section in a message - ecCodes BUFR FAQ](#) — In a BUFR message it is possible to have an optional section 2 to be used as local section for information that is relevant for the processing and archiving systems used by a processing centre. Adding or removing a local section from a BUFR message is very easy with ecCodes by setting the key "section2Present"
- [Local configuration - ecCodes BUFR FAQ](#) — Users can define their own local BUFR tables to adapt ecCodes to their own needs e.g. to use a local descriptor.
- [How do I convert BUFR edition 3 to 4 - ecCodes BUFR FAQ](#) — To convert a message the user simply sets `edition=4` and the message will be automatically converted to a valid BUFR edition 4 message.
- [How do I change the data structure in a message - ecCodes BUFR FAQ](#) — A BUFR message is made of some headers and a data section. The structure of the data section is defined in the headers by a sequence of BUFR descriptors, which are sequences of six digit numbers having a particular meaning (see [BUFR tables](#)). To modify this sequence of descriptors ecCodes provides the key `unexpandedDescriptors`.
- [Encoding of Delayed Replications - ecCodes BUFR FAQ](#) — A delayed replication is a way of replicating n times a block of data in a BUFR message by expressing the value of n in the data section itself as opposed to repeating the blocks statically in the definition of the structure. This is possible because the structure of the message is defined in section 3 through the sequence of `unexpandedDescriptors` while the data themselves are encoded in section 4 (data section) according to the structure defined in section 3.
- [grib_filter/bufr_filter: How do I control the format of array printing - ecCodes GRIB FAQ](#) — One can add several 'modifiers' to a key in square brackets to control number of columns, separator string and format
- [Setting keys by rank - ecCodes BUFR FAQ](#) — In a BUFR message the same key can appear several times in different positions of the message. It is possible to set a key by rank.