

# BUFR structure - ecCodes BUFR FAQ

BUFR messages are composed of *sections*. The sections encode the *metadata* as well as the *data* itself.

A BUFR message is composed of six sections, numbered zero through five.

- Sections 0, 1 and 5 contain static metadata, mostly for message identification.
- Section 2 is optional; if used, it may contain arbitrary data in any form wished for by the creator of the message (this is only advisable for local use).
- Section 3 contains a sequence of so-called *descriptors* that define the form and contents of the BUFR data product.
- Section 4 is a bit-stream containing the message's core data values as laid out by Section 3



The metadata (the headers) which describe the structure of the data are always available. Therefore the keys in these sections are accessible as soon as you get a message handle (e.g. via the Python function `codes_buf_r_new_from_file()`)  
To access the data section, you need a special directive which instructs ecCodes to do the actual unpacking and decoding of all the data elements. This is done by setting the special key **unpack** to 1.

[Here](#) is an example of decoding only the header of a BUFR message.  
And [here](#) we have an example of decoding all the data (header + data).

A good way of seeing the contents of two parts is via the [BUFR Validator](#). Load your favourite BUFR file there and click Validate and you will see two top-level entries: Header and Data. Expand each one to reveal the ecCodes BUFR keys and their respective values. To decode the keys under Data requires the setting of the above-mentioned "unpack" key.

For further details, please consult the eLearning resource [Introduction to BUFR decoding with ecCodes](#) (login needed to access content on Learning Platform).

## Related articles

- [Dump instructions to decode the input - ecCodes BUFR FAQ](#)
- [Performance improvement by skipping some keys - ecCodes BUFR FAQ](#)
- [What is a subset - ecCodes BUFR FAQ](#)
- [Missing values - ecCodes BUFR FAQ](#)