ecCodes BUFR Command-Line Interface (CLI) Tools

Chris Kelly

Software Engineer, Observations Team

christopher.kelly@ecmwf.int

CLI tools vs. Python interface

- ecCodes BUFR includes both CLI tools & a Python interface

Command-Line Interface	Python Interface
• Rapid file inspection and filtering	• Complex data integration and analysis
• On-the-fly modifications	• Complete observation processing pipeline
• Batch quality checks	• Sophisticated quality control
• Lightweight regression testing	• Reproducible scientific workflows

Key functions of CLI tools

- Examine BUFR file/message structure and contents

  → **bufr_count**, **bufr_ls**, **bufr_dump**

- Copy BUFR file content

  → **bufr_copy**

- Simple data retrievals and modifications

  → **bufr_get, bufr_set**

- Advanced data retrievals and modifications

  → **bufr_filter**

- Compare BUFR messages contained in two files

  → **bufr_compare**

Example data access

- Global Upper Air Radiosonde BUFR archive from ECMWF

  → **Search:** gov.noaa.ncdc:C01549

  → **Download:** Upperair-BUFR_ECMWF_d202201_c20250206.tar.gz (*1.7 GB*)

- Citation: *Xungang Yin; DOC/NOAA/NESDIS/NCEI > National Centers for Environmental Information, NESDIS, NOAA, U.S. Department of Commerce*

Examine BUFR file/message structure and contents (1/4)

- bufr_count: Print the number of messages contained in a BUFR file.

```
$ bufr_count BTEF00012022010100.DC1.DAT

1519
```

- bufr_ls: List select contents from every message contained in a BUFR file.

```
$ bufr_ls BTEF00012022010100.DC1.DAT

BTEF00012022010100.DC1.DAT

centre                      masterTablesVersionNumber  localTablesVersionNumber (→ continued)
ecmf                        27                         0
ecmf                        27
(↑ continued ↓)
ecmf                        31                         0
ecmf                        21                         0                          1519 of 1519
messages in BTEF00012022010100.DC1.DAT

1519 of 1519 total messages in 1 files
```

Examine BUFR file/message structure and contents (2/4)

- bufr_dump: Print the full content of a BUFR file.

```
$ bufr_dump -p -w count=1 BTEF00012022010100.DC1.DAT

delayedDescriptorReplicationFactor= {0}
extendedDelayedDescriptorReplicationFactor= {75}
edition=3
masterTableNumber=0
bufrHeaderSubCentre=0
bufrHeaderCentre=98
updateSequenceNumber=0
dataCategory=2
dataSubCategory=0
masterTablesVersionNumber=27
localTablesVersionNumber=0
typicalYearOfCentury=22
(↑ continued ↓)
observedData=1
compressedData=0
----------------------------------
continued ↓)
```

- -p: plan/flat dump
- -w: where clause
- count: bufr message number

Section 0 (indicator section),
Section 1 (identification section) &
Section 2 (optional section).

Examine BUFR file/message structure and contents (3/4)

- bufr_dump: Print the full content of a BUFR file.

```
$ bufr_dump -p -w count=1 BTEF00012022010100.DC1.DAT

(↑ continued)
unexpandedDescriptors={
      301128, 007007, 002102, 201133, 025065, 025066, 201000,
      309052, 205011, 005001, 006001, 007001 }
continued ↓)
```

Section 3 (data description section).

Unravelling unexpanded descriptors

```
→ sequence.def

"301128" = [  001081, 001082, 001083, 001095, 002015, 002016, 002017, 002066, 002067, 002080, 002081,
002082, 002083, 002084, 002085, 002086, 002095, 002096, 002097, 002103, 002191, 025061, 035035 ]

        → element.table

        001081|radiosondeSerialNumber|string|RADIOSONDE SERIAL NUMBER|CCITT IA5|0|0|160|Character|0|20
        001082|radiosondeAscensionNumber|long|RADIOSONDE ASCENSION NUMBER|Numeric|0|0|14|Numeric|0|4
        001083|radiosondeReleaseNumber|long|RADIOSONDE RELEASE NUMBER|Numeric|0|0|3|Numeric|0|1
        …
        002191|geopotentialHeightCalculation|table|GEOPOTENTIAL HEIGHT CALCULATION|CODE TABLE|0|0|4|CODE TABLE|0|2
        025061|softwareVersionNumber|string|SOFTWARE IDENTIFICATION AND VERSION NUMBER|CCITT IA5|0|0|96|Character|0|12
        035035|reasonForTermination|table|REASON FOR TERMINATION|CODE TABLE|0|0|5|CODE TABLE|0|2
```

Examine BUFR file/message structure and contents (4/4)

- bufr_dump: Print the full content of a BUFR file.

```
$ bufr_dump -p -w count=1 BTEF00012022010100.DC1.DAT
(↑ continued)
radiosondeSerialNumber="2245411/82919"
(↑ continued ↓)
#75#timePeriod=4150
#75#extendedVerticalSoundingSignificance=65536
#75#pressure=10000
#75#nonCoordinateGeopotentialHeight=15300
#75#latitudeDisplacement=0.27636
#75#longitudeDisplacement=-0.13429
#75#airTemperature=215.95
#75#dewpointTemperature=209.75
#75#windDirection=195
#75#windSpeed=19
text="61616 20312"
#2#latitude=66.77
#2#longitude=123.4
heightOfStation=92
```

The BUFR needs to be unpacked to access the following keys.

Section 4 (data section) & Section 5 (end section).

Rank numbers (e.g. #75#) are used to maintain uniquely identifiable BUFR keys when a descriptor needs to be reused (e.g. sounding profile levels).

Copy BUFR file content

- bufr_copy: Copy contents according to specified criteria from an existing BUFR file into a new BUFR file.

```
$ bufr_copy -w count=1 BTEF00012022010100.DC1.DAT BTEF00012022010100_msg1.DC1.DAT
```

This example extracts the first message from the original BUFR and copies it into a new, single-message BUFR.

- -w: where clause
- count: bufr message number

Simple data retrievals and modifications

- bufr_get: Print specified contents from BUFR messages.

```
$ bufr_get -p typicalYearOfCentury,typicalMonth,typicalDay,typicalHour,typicalMinute
BTEF00012022010100.DC1.DAT

22 1 1 0 24
22 1 1 0 24
(↑ continued ↓)
21 12 31 21 57
21 12 31 21 50
```

- -p: keys to print
- Multiple keys may be specified in a list, using a comma without spaces.
- Only keys that don't require the BUFR to be unpacked may be retrieved (Sections 0-2).

- bufr_set: Sets or changes the value of a key in BUFR messages.

```
$ bufr_set -p bufrHeaderCentre -s bufrHeaderCentre=254 BTEF00012022010100.DC1.DAT
BTEF00012022010100_bHC254.DC1.DAT
```

- -p: keys to print
- -s: keys/values to set
- Only keys that don't require the BUFR to be unpacked may be modified (Sections 0-2).

Advanced data retrievals and modifications

- bufr_filter: Transform BUFR messages using specified rules defined in a ‘rules file’.

```
$ bufr_filter -o BTEF00012022010100_statnum.DC1.DAT rules.flt BTEF00012022010100.DC1.DAT
```

```
→ rules.flt
set unpack=1;
set stationNumber=123;
set pack=1;
write;
```

This example sets the key ‘stationNumber’ equal to 123 in every message.

- -o: output file
- rules.flt: rules file
- Keys that require the BUFR to be unpacked may be also modified.

```
$ bufr_filter -o BTEF00012022010100_latlon.DC1.DAT rules.flt BTEF00012022010100.DC1.DAT
```

```
→ rules.flt
if ((localLatitude >= 30 && localLatitude <= 60) &&
    (localLongitude >= 0 && localLongitude <= 90)) {
    write;
}
```

This example filters every message by a specified latitude-longitude range.

Compare BUFR messages contained in two files

- bufr_compare: Print the differences between messages contained in two BUFR files.

```
$ bufr_compare BTEF00012022010100_statnum.DC1.DAT BTEF00012022010100.DC1.DAT

== 1 == DIFFERENCE == long [stationNumber]: [123] != [343]
```

This example compares the output of our first bufr_filter to the original BUFR.

Additional options

- There are many more additional options that may be used with the CLI tools.
- To see what's available, use '-h'.

E.g.

```
$ bufr_dump -h

NAME    bufr_dump
DESCRIPTION
        Dump the content of a BUFR file in different formats.
(↑ continued ↓)
        -O      Octet mode. WMO documentation style dump.
        -p      Plain/Flat dump (key=value format).
        -d      Dump the expanded descriptors.
        -t      Print type information.
        -f      Force. Force the execution not to fail on error.
        -H      Print octet content in hexadecimal format.
        -a      Dump aliases.
(continued ↓)
```

Test your knowledge (Qs)

→ **Search:** gov.noaa.ncdc:C01549
→ **Download:** Upperair-BUFR_ECMWF_d202201_c20250206.tar.gz

1. Write a bufr_set to make the qualityControl key equal to 100.

```
$
```

2. Write a bufr_get to check that our lat-lon bufr_filter worked as expected.

```
$
```

3. Calculate how many BUFR messages were removed by our lat-lon filter.

```
$
```

Test your knowledge (Q&As)

1. Write a bufr_set to make the qualityControl key equal to 100.

```
$ bufr_set -p qualityControl -s qualityControl=100 BTEF00012022010100.DC1.DAT
BTEF00012022010100_qc100.DC1.DAT
```

2. Write a bufr_get to check that our lat-lon bufr_filter worked as expected.

```
$ bufr_get -p localLatitude,localLongitude BTEF00012022010100_latlon.DC1.DAT

39.3443 8.96427
(↑ continued ↓)
50.5618 10.3771
```

3. Calculate how many BUFR messages were removed by our lat-lon filter.

```
$ bufr_count BTEF00012022010100_latlon.DC1.DAT

246
```

1519 – 246 = 1273