

BUFR decoding

Dominique Lucas
User Support

Content

- What is BUFR
- BUFR tools
- BUFR format
- BUFR decoding
- Practical examples

What is BUFR

- Binary representation of meteorological data or **B**inary **U**niversal **F**orm for data **R**epresentation.
- Continuous bit stream made of sequence of octets.
- Table driven code.
- Self descriptive code.
- Machine independent.
- Compression available for improved transmission speed.

BUFR tools – data validation

The screenshot shows the ECMWF BUFR validator web interface. The browser address bar is `apps.ecmwf.int/codes/bufr/validator/`. The page title is "BUFR validator". A navigation bar at the top contains the ECMWF logo, "About", "Forecasts", "Computing", "Research", "Learning", a search box, and the user name "Dominique Lucas".

On the left, there is a "Manuals" section with links to "ecCodes Documentation" and "WMO Manual".

The main content area has the heading "BUFR validator" and a description: "The purpose of the validator is to verify the compliance of data in BUFR format with the specifications in the WMO Manual on Codes." Below this is a file upload section with a "Browse..." button, the text "No file selected.", and a "Validate" button. A note states "The file size is limited to 2 megabytes." A success message reads: "The file bufr_file has been validated. It contains 1 message." Below the message, there is a tree view with "Header" and "Data" sections. The "Data" section is expanded to show the following fields:

subsetNumber:	1	
blockNumber:	3	[percentConfidence = 70 %]
stationNumber:	955	[percentConfidence = 70 %]
stationType:	1	[percentConfidence = 70 %]
year:	2011 a	[percentConfidence = 70 %]
month:	3 mon	[percentConfidence = 70 %]
day:	4 d	[percentConfidence = 70 %]
hour:	12 h	[percentConfidence = 70 %]
minute:	0 min	[percentConfidence = 70 %]

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the time "12:19 PM 2/20/2016".

Using eccodes

BUFR data examiner - metview

The screenshot displays the BUFR data examiner application. At the top, there is a menu bar (File, Edit, View, Profiles, Help) and a toolbar with icons for file operations. The 'Key profile' is set to 'mv System::Default'. The main window shows the file 'IUSD40_OKLI.bufr' with permissions '-rw-r-----', owner 'usl', group 'us', size '8.0KB', and modification date '2011-03-07 14:39'. It also indicates 'Total number of messages: 4'. Below this, there are input fields for 'Go to message: 3' and 'Go to subset: 1' (with a note '(Number of subsets: 1)').

Index	△	Typ	Sut	C	Ssc	Date	Time	Lat1
1		2	0	89	1	2007-11-20	18:00	N/A
2		2	0	89	1	2007-11-20	12:00	N/A
3		2	0	89	1	2007-11-20	06:00	N/A
4		2	0	89	1	2007-11-20	00:00	N/A

A yellow box highlights the text 'Using BUFRDC' in the top right corner. A blue box highlights the command '\$ metview -e bufr ~trx/bufr_decode/bufr_file' in the bottom left area. On the right side, a detailed view of 'Section 0-3' is shown, with tabs for 'Data' and 'Data, bitmaps expanded'. The section details are as follows:

Section	Name	Value
Section 0	LENGTH OF SECTION 0 (BYTES)	8
	TOTAL LENGTH OF BUFR MESSAGE (BYTES)	1286
	BUFR EDITION NUMBER	3
Section 1	LENGTH OF SECTION 1 (BYTES)	18
	BUFR EDITION NUMBER	3
	ORIGINATING SUB-CENTRE	0
	ORIGINATING CENTRE	89
	UPDATE SEQUENCE NUMBER	0
	FLAG (PRESENCE OF SECTION 2)	0
	BUFR MESSAGE TYPE	2
	BUFR MESSAGE SUBTYPE	0
	VERSION NUMBER OF LOCAL TABLE	0
	YEAR	7

The status bar at the bottom left shows 'Status: OK'.

BUFR tools - ecCodes

\$ bufr_ls

List content of BUFR files printing values of some keys.
It does not fail when a key is not found.

\$ bufr_dump

Dump the content of a BUFR file in different formats.

\$ bufr_compare

Compare BUFR messages contained in two files.

\$ bufr_filter

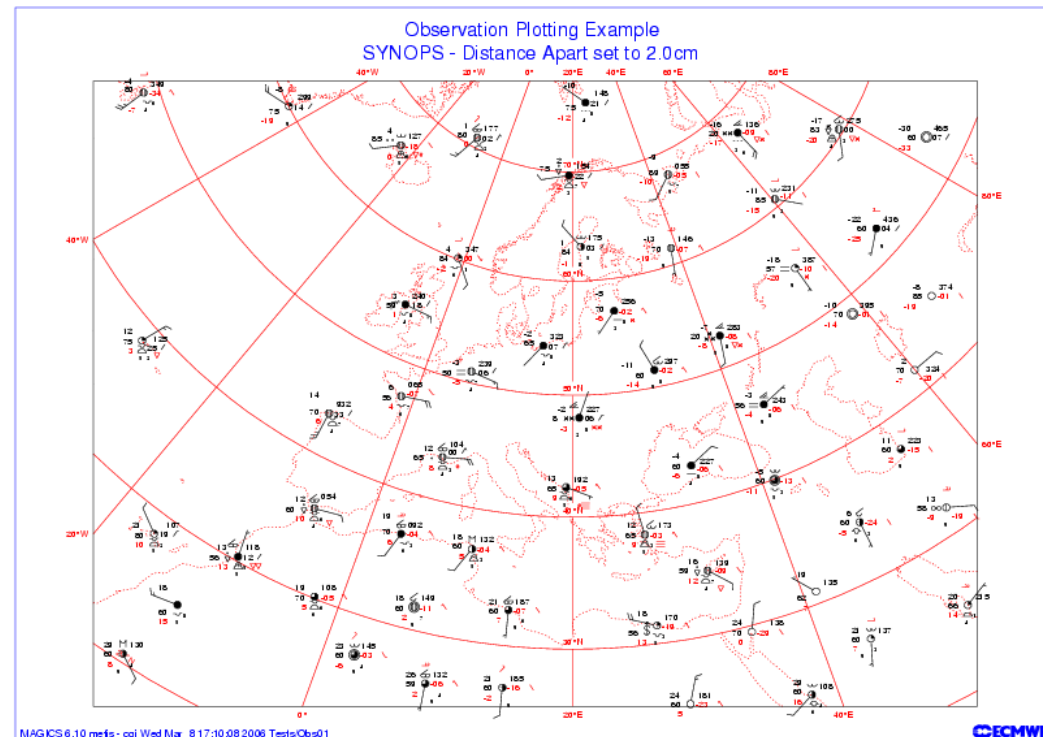
Apply the rules defined in rules_file to each BUFR message
in the BUFR files provided as arguments.

BUFR format

- Section 0: Indicator section
- Section 1: Identification section
- Section 2: Optional section
- Section 3: Data description section
- Section 4: Data section
- Section 5: End section
 - All sections are padded with “0”s if needed to occupy even number of octets.

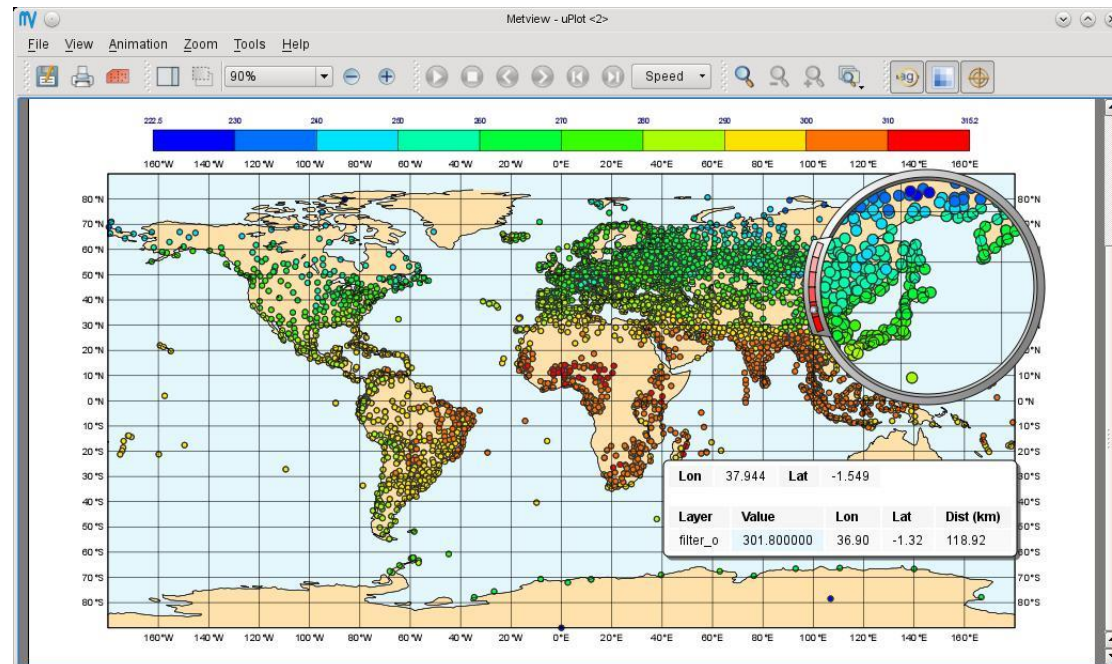
Section 0 - Indication section

- 4 characters 'BUFR'
- Length of message
- Edition Number



Section 1 - Identification section

- Originating Centre
- Data Category (Table A) and sub-category
- Version number of tables
- Date and time



Section 2 – Optional section

- At ECMWF, section 2 contains data used by MARS.

File Edit View Profiles Help
Key profile: System:Default

File: /home/ectrain/trx/bufr_decode/bufr_file
Permissions: -rw-r--r-- Owner: trx Group: ectrain Size: 4.0KB Modified: 2011-03-07 14:34
Total number of messages: 1

Go to message: 1 Go to subset: 1 (Number of subsets: 1)

Section 0-3 Data Data, bitmaps expanded

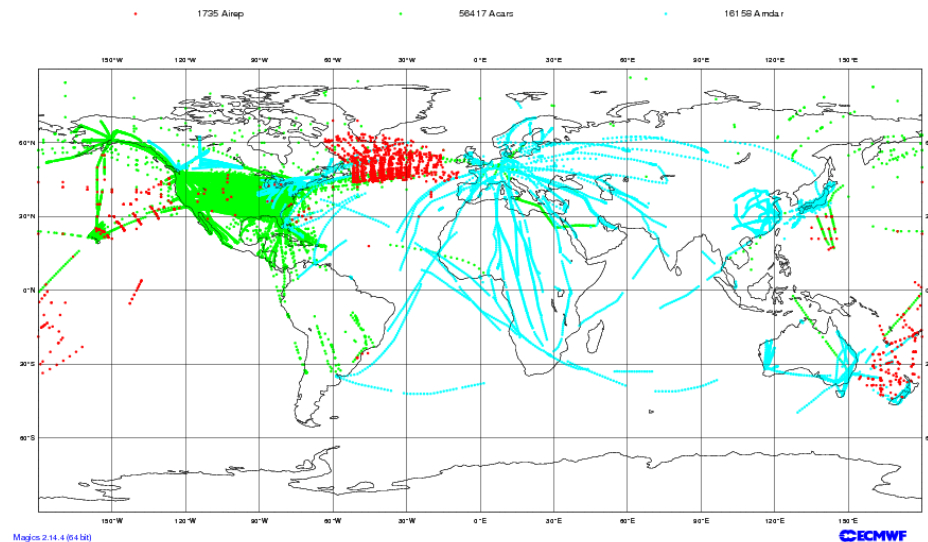
Section	Name	Value
Section 2		
	LENGTH OF SECTION 2	52
	REPORT DATA BASE	KEY
	RDB DATA TYPE	1
	RDB DATA SUBTYPE	1
	YEAR	2011
	MONTH	3
	DAY	4
	HOUR	12
	MINUTE	0
	SECOND	0
	LATITUDE 1	51.85
	LONGITUDE 1	-8.48
	IDENTIFER	03955
	TOTAL BUFR MESSAGE LENGTH	220
	DAY (RDB INSERTION)	4
	HOUR (RDB INSERTION)	12
	MINUTE (RDB INSERTION)	15
	SECOND (RDB INSERTION)	4
	DAY (MDB ARRIVAL)	4
	HOUR (MDB ARRIVAL)	12
	MINUTE (MDB ARRIVAL)	14
	SECOND (MDB ARRIVAL)	11
	CORRECTION NUMBER	0
	PART OF MESSAGE	1
	CORRECTION NUMBER	0
	PART OF MESSAGE	0
	CORRECTION NUMBER	0

Status: OK

Section 3 – Data description section

- Number of data subsets
- Flag for compression
- Data descriptors

ECMWF Data Coverage (All obs DA) - Aircraft
04/Mar/2013; 00 UTC
Total number of obs = 74310



Section 3 – Data descriptors

F type	X category	Y entry
2 bits	6 bits	8 bits

- F = 0 Element Descriptor – Bufr table B
- F = 1 Replication descriptor
 - X = number of descriptors to repeat
 - Y = number of times the descriptors are repeated
- F = 2 Operator Descriptor – “Bufr table C”
- F = 3 Sequence Descriptor – Bufr table D

Section 4 - Data section

- Binary data

Section 5 - End section

- 4 digits '7777'

BUFR Tables

- Table A - Data category
- Table B - Classification of elements
- Code and flag table
- Table C - Data Descriptor operators
- Table D - List of common sequences

http://www.wmo.int/pages/prog/www/WMOCodes/WMO306_v12/LatestVERSION/LatestVERSION.html

Table A - Data category

- Used in the Section 1 (element 9) of the BUFR message
- Example:

<u>Code figure</u>	<u>Meaning</u>
0	Surface data - land
1	Surface data – sea
2	Vertical soundings (not satellite) ...
31	Oceanographic data

Table B - Classification of elements

Element Name	Unit	Scale	Reference	#bits
• 005001 Latitude (high accuracy)	Degree	5	-9000000	25
• 007003 Geopotential	m**2/s**2	-1	-400	17
• 002019 Satellite instruments	Code table	0	0	11
• 008001 Vertical sounding signifi	Flag Table	0	0	7
• 001006 Aircraft flight number	CCITTIA5	0	0	64
• 011012 Wind speed at 10m	m/s	1	0	12

- (obs. * 10**scale – Reference) is encoded into #bits bits
- For coded or flagged values, the element descriptor indicates the number of the table describing the codes/flags.

- 0 - Table B entry
 - 05 - Location (horizontal 1) class
 - 01 - Identification
 - 08 - Significance qualifiers

Code and flag tables

- 0 20 003 – Present Weather

<u>Code figure</u>	<u>Meaning</u>
0	Cloud development not observed or not observable
1	Clouds generally dissolving or becoming less developed
...	
10	Mist
11	Patches of shallow fog or ice fog
...	
61	Rain, not freezing, continuous; slight at time of obs.
...	
171	Snow, slight
172	Snow, moderate
173	Snow, heavy
...	
511	Missing

Table C - Data Descriptor operators

- 201yyy - Change data width
- 202yyy - Change scale
- 203yyy - Change reference value

- 222000 - Quality information

This table is internal to the BUFR software.

See:

http://www.wmo.int/pages/prog/www/WMOCodes/WMO306_v12/LatestVERSION/LatestVERSION.html

Table D - List of common sequence

- Table D can contain sequences of table B entries, Table D entries and Operators. It is not needed but saves a lot of space.

– 301027	301001	WMO block and station
	002011	Radiosonde type
	002012	Radiosonde computational method
	301011	Date
	301012	Time
	301022	Lat/Long and station height

See [Bufr templates](#) giving common sequences for some observations.

BUFR decoding - ecCodes

- ecCodes: GRIB and BUFR decoding software
 - Fortran 90, C and python interfaces.
 - BUFR tools.
 - The same API for decoding GRIB and BUFR data.
- Beta-released in 2015, fully released in 2016.
- ecCodes has to be loaded with 'module':
`$ module swap grib_api eccodes`

<https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>

BUFR decoding - ecCodes

Sample fortran code

```
call codes_open_file(ifile,'syno_multi.bufr','r')
```

I/O routines

```
! the first bufr message is loaded from file
```

```
! ibufr is the bufr id to be used in subsequent calls
```

```
call codes_bufr_new_from_file(ifile,ibufr,iret)
```

...

```
! we need to instruct ecCodes to expand all the descriptors
```

```
! i.e. unpack the data values
```

```
call codes_set(ibufr,"unpack",1);
```

decoding routine

```
call codes_get(ibufr,'typicalDate',typicalDate)
```

```
write(*,*) ' typicalDate:',typicalDate
```

BUFR decoding - ecCodes

- Hints for decoding:
 - Attributes of a key: operator ‘->’
"key" : "blockNumber",
"code" : "001001"
‘blockNumber->code’
 - Accessing keys by rank: operator ‘#’
Subset number or replicated descriptors
‘#2#pressure’

BUFR decoding - ecCodes

- Hints for encoding
 - Cloning a message:
`call codes_clone(ibufr, obufr)`
 - Using a sample message:
`call codes_buf_r_new_from_sample(ibufr, 'BUFR4')`
See 'codes_info' for samples directory.
 - Using 'buf_r_dump -E<format>':
`buf_r_dump -Epython synop.buf_r > synop.py`

Note: Bufr messages created with ecCodes will be syntactically correct, but may be semantically incorrect.

BUFR decoding - BUFRDC

- Included within EMOSLIB, loadable with 'modules':
\$ module load emos
- Old Fortran 77 code, difficult to maintain.
- Fortran API.
- Oriented to the sections of the BUFR format.
- One main routine to decode, called BUFREX.
- Indirect access to the meteorological data via the expanded data descriptors.
- Software no longer developed. Bufr tables are updated.

<https://software.ecmwf.int/wiki/display/BUFR/BUFRDC+Home>

BUFR Tables

- ecCodes
 - The BUFR tables are part of the definition files of ecCodes (see ‘codes_info’) and have been reorganized, e.g.
 - [eccodes/definitions/bufr/tables/0/wmo/6/codetables](#)
 - [eccodes/definitions/bufr/tables/0/wmo/6/element.table](#)
 - [eccodes/definitions/bufr/tables/0/wmo/6/sequence.def](#)
 - One can access customized tables by defining the env. variable `ECCODES_DEFINITION_PATH`.
- BUFRDC
 - text files name like, e.g., B00000000000098013001.TXT included with the distribution of EMSOLIB.
 - One can point to a customised version of these tables using the env. variable called `BUFR_TABLES`.

Where to find more about BUFR

- BUFR format

<http://www.wmo.int/pages/prog/www/WMOCodes.html>

- ecCodes software

<https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>

- ecCodes BUFR API examples

<https://software.ecmwf.int/wiki/display/ECC/BUFR+examples>

- BUFRDC software

<https://software.ecmwf.int/wiki/display/BUFR/BUFRDC+Home>

Practical examples

- Copy files across ...

```
$ cd $SCRATCH
```

```
$ tar xvf ~trx/bufr_decode/practicals.tar
```

```
$ cd bufr_decode
```

- Familiarise yourself with the bufr examiner in metview, e.g.

```
$ metview -e BUFR synop.bufr
```

- You can now move on to using ecCodes.

```
$ module swap grib_api eccodes # not needed with training IDs.
```

Practical examples

- Use the 'bufr' tools (bufr_ls and bufr_dump) to list the content of the file bufr_file. E.g.:
\$ bufr_dump -O synop.bufr
\$ bufr_dump -ja synop.bufr
- Other bufr files are also available. Some are compressed, some include multiple subsets.
- The source code file bf_ecodes_extract.f90 uses ecCodes to extract some meteorological data from a BUFR message:
\$ make bf_ecodes_extract
\$./bf_ecodes_extract

Practical examples

- The python script `bf_ecodes_extract.py` does the same as `bf_ecodes_extract.90` :

```
$ python bf_ecodes_extract.py
```

- The source code file in `bfextract.f90` will do the same as `bf_ecodes_extract.90`, but using BUFRDC:

```
$ make bfextract
```

```
$ ./bfextract
```

- You can use `'bufr_dump [-D|E]<format>'` to generate a sample code to decode or encode a BUFR message:

```
$ bufr_dump -Efortran temp.bufr > create_temp.f90
```

```
$ gfortran -o create_temp create_temp.f90 $ECCODES_INCLUDE $ECCODES_LIB
```

```
$ create_temp
```