

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 interface. The browser address bar is `apps.ecmwf.int/codes/bufr/validator/`. The page title is "BUFR validator". A navigation bar includes "About", "Forecasts", "Computing", "Research", "Learning", a search box, and a user profile "Dominique Lucas".

Manuals

- [ecCodes Documentation](#)
- [WMO Manual](#)

The purpose of the validator is to verify the compliance of data in BUFR format with the specifications in the [WMO Manual on Codes](#).

Use the file box to upload your BUFR file

No file selected.

The file size is limited to 2 megabytes.

The file bufr_file has been validated.
It contains 1 message.

► Header

▼ Data

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 %]

Windows taskbar: Search the web and Windows, system tray with time 12:19 PM 2/20/2016.

Using eccodes

BUFR data examiner - metview

File Edit View Profiles Help

Key profile: mv System::Default

File: IUSD40_OKLI.bufr
Permissions: -rw-r----- Owner: us1 Group: us Size: 8.0KB Modified: 2011-03-07 14:39
Total number of messages: 4

Using BUFRDC

Go to message: 3 Go to subset: 1 (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

\$ metview -e bufr ~trx/bufr_decode/bufr_file

Status: OK

BUFR tools - ecCodes

\$ bufr_ls

DESCRIPTION

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

\$ bufr_dump

DESCRIPTION

Dump the content of a BUFR file in different formats.

\$ bufr_compare

DESCRIPTION

Compare BUFR messages contained in two files.

\$ bufr_filter

DESCRIPTION

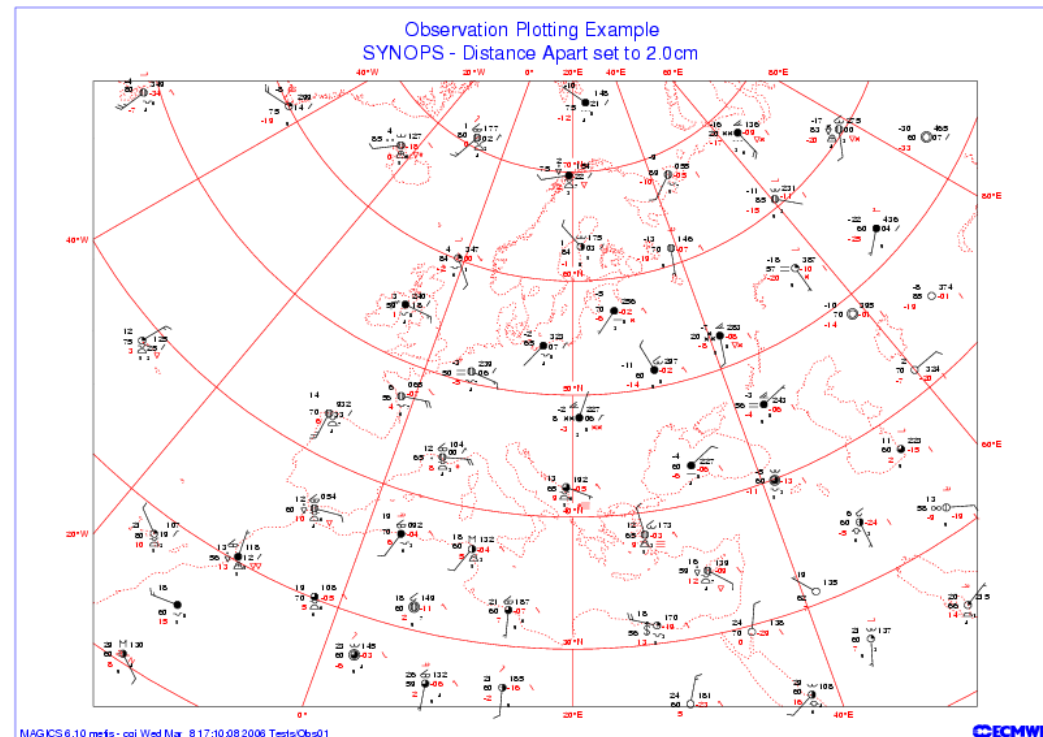
Apply the rules defined in rules_file to each BUFR message
in the BUFR files provided as arguments.

BUFR format

- Indicator section
- Identification section
- Optional section
- Data description section
- Data section
- End section
 - All sections are padded with “0”s if needed to occupy even number of octets.

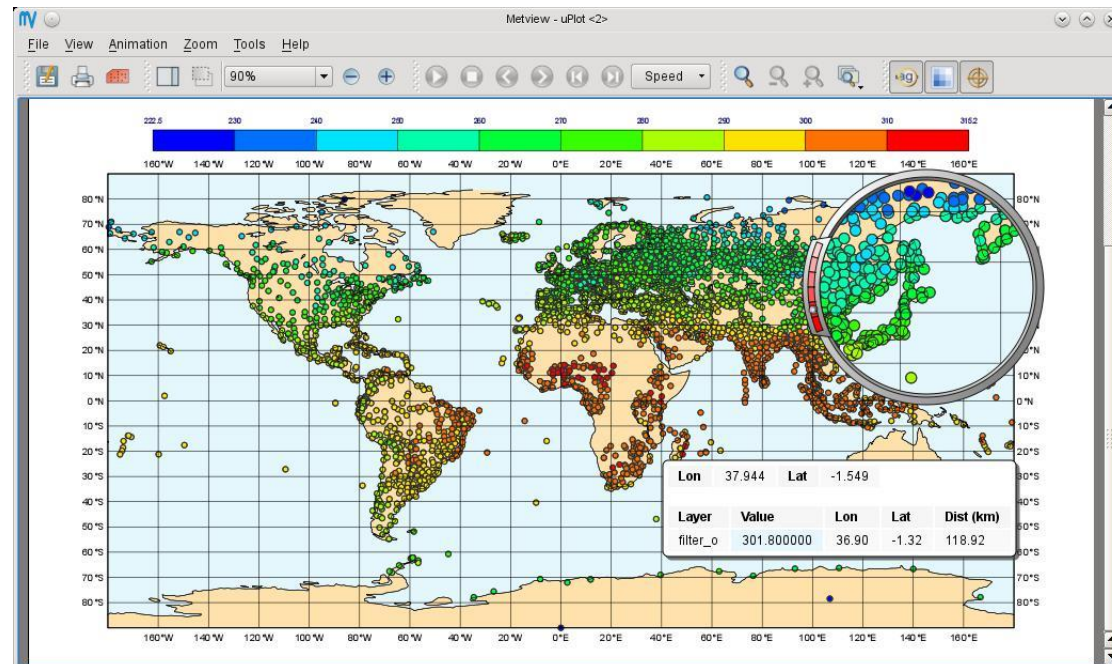
Section 0 - Indication section

- 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: mv 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

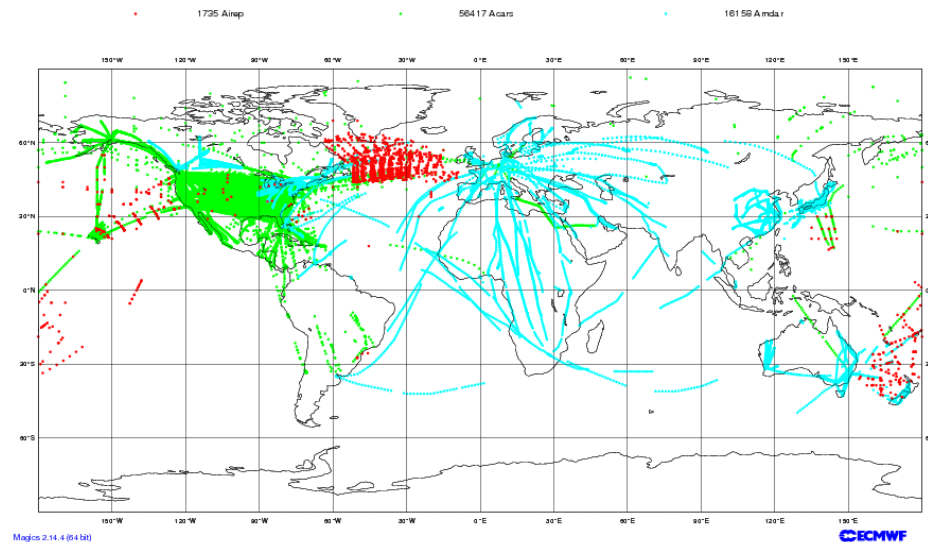
Section	Name	Value
Section 2		
	LENGTH OF SECTION 2	52
	REPORT DATA BASE	KEY
	RDB DATA TYPE	1
	RDB DATA SUBTYPE	1
	YEAR	2011
	MONTH	3
	DAY	4
	HOUR	12
	MINUTE	0
	SECOND	0
	LATTITUDE 1	51.85
	LONGITUDE 1	-8.48
	IDENTIFER	03955
	TOTAL BUFR MESSAGE LENGTH	220
	DAY (RDB INSERTION)	4
	HOUR (RDB INSERTION)	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 – “Internal table C”
- F = 3 Sequence Descriptor – Bufr table D

“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

Section 4 - Data section

- Binary data

Section 5 - End section

- 4 digits '7777'

BUFR Tables

- Table A - Data category
- Table B - Classification of elements
- Table C - Code and flag table (*)
- Table D - List of common sequences

http://www.wmo.int/pages/prog/www/WMOCodes/WMO306_vI2/LatestVERSION/LatestVERSION.html

(*) Not to mix with the table C with 'data descriptor operators' mentioned before.

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

Table C – 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 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

BUFR decoding - BUFRDC

- Official BUFR decoding software.
- 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.
- For a flavour of BUFRDC, see the practical session.

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

BUFR decoding - ecCodes

- ecCodes: GRIB and BUFR decoding software
 - rewrite of BUFRDC.
 - Fortran 90, C and python interfaces.
 - BUFR tools.
 - The same API for decoding GRIB and BUFR data.
- Beta-released in 2015, to be fully released this year.
- ecCodes has to be loaded with modules:
 - \$ module unload grib_api
 - \$ module load 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 Tables

- 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.
- 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.

Where to find more about BUFR

- BUFR format
<http://www.wmo.int/pages/prog/www/WMOCodes.html>
- BUFRDC software
<https://software.ecmwf.int/wiki/display/BUFR/BUFRDC+Home>
- ecCodes software
<https://software.ecmwf.int/wiki/display/ECC/ecCodes+Home>
- ecCodes BUFR API examples
<https://software.ecmwf.int/wiki/display/ECC/BUFR+examples>

Practical examples – to be run on ecgate

- 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 bufr_file
- The source code file 'bfdemo.f90' decodes a BUFR Synop message using BUFRDC and is available together with the Makefile. The file 'bfdemo.c' contains C code.
\$ make bfdemo # or make bfdemoc
\$./bfdemo # or ./bfdemoc

Practical examples – to be run on ecgate

- Can you adapt the above script/code to write part of the observation data out to a file, e.g.

```
! 001001 WMO BLOCK NUMBER
! 001002 WMO STATION NUMBER
! 004001 YEAR
! 004002 MONTH
! 004003 DAY
! 004004 HOUR
! 004005 MINUTE
! 005001 LATITUDE (HIGH ACCURACY)
! 006001 LONGITUDE (HIGH ACCURACY)
! 010004 PRESSURE
! 011011 WIND DIRECTION AT 10 M
! 011012 WIND SPEED AT 10 M
```

The source code file is `bfextract.f90` shows an example to do this using BUFRDC:

```
$ make bfextract
```

```
$ ./bfextract
```

Practical examples – to be run on ecgate

- You can now move on to using ecCodes.

```
$ module unload grib_api
```

```
$ module load eccodes
```

Use the 'bufr' tools (bufr_ls and bufr_dump) to list the content of the file bufr_file. E.g.:

```
$ bufr_dump -O bufr_file
```

The source code file bf_ecodes_extract.f90 uses ecCodes to do the same as bfextract.f90

```
$ make bf_eccodes_extract
```

```
$ ./bf_eccodes_extract
```