EUMETSAT

BUFR Formatting Software Specification, Design and User Documentation

Doc.No. : EUM/JAS/TEN/07/0012
Issue : v10
Date : 1 February 2010

EUMETSAT
Am Kavalleriesand 31, D-64295 Darmstadt, Germany
Tel: +49 6151 807-7
Fax: +49 6151 807 555 Telex: 419 320 metsat d
http://www.eumetsat.int

Document Signature Table

	Name	Function	Signature	Date
Prepared by:	Simon Elliott	Product Implementation Manager		2/2/10
Reviewed by:	D. Faucher	QAD Engineer		2/2/10
	H. Bonekamp	Jason 2 Mission Scientist		2/2/10
	R. Zarza	System Engineer		2/2/10
Approved by:	S. Dieterle	Jason-2 Project Manager		2.2.2010

Distribution List

Distribution list	
Name	**No. of Copies**
MED/RZ, MED/C/MTa, MED/C/VPr, QAD/DF, CCD/SDI, MET/HGB, MOD/SSE, MOD/JF	Electronic link via documentation management system (Hummingbird DM)

Document Change Record

Issue / Revision	*Date*	*DCN. No*	*Changed Pages / Paragraphs*
v1	April 2007		Drafted for Operation Preparation Meeting and initial implementation
v2	27 Dec. 07		Updated following regression testing
v3	14 March 2008		Update to reflect version supporting addition requirements from NOAA side (two files created and command line options)
v4	27 March 2008		Updated to reflect use of BUFR master table version 13 and local table version 0. Clarify scope of data checking in Section 3, in response to JASON-2 Internal ORR RID number 13.
v5	25 June 2008		Update FIST version to 1.3. Add Appendix explaining mapping between flags in netCDF and BUFR.
v6	19 August 2008		Update FIST version to 1.4. Clarify minor design change to deal with negative temperatures (AR 86).
v7	14 November 2008		Update FIST version to 1.5. Change applicable version of AD8, SALP Products Specification (ECP 5).
v8	26 November 2008		Update FIST version to 1.6. Handle revised processing centre names in netCDF global attributes (ECP 5).
v9	26 June 2009		Update FIST version to 1.7. Corrections for sundry anomalies (AR 131, 134 and 136) and change to GTS file names (AR 130).
v9A	6 July 2009		Remove correction of AR 130 (country code in GTS file names) from FIST version 1.7.
v9B	7 July 2009		Update signature list.
v10	1 February 2010		Update FIST version to 1.8. Corrections for change to GTS file names (AR 130), updated to BUFR encoding library (AR 143) and

			revision of encoding to include additional variables and new BUFR sequence (AR 160).

Table of Contents

1 INTRODUCTION

1.1 Purpose and Scope

The purpose of this document is to present the specification, design and minimum useful documentation for the development, installation and test activities related to the piece of software developed for converting netCDF formatted JASON2 OGDR files into BUFR.

1.2 Applicable Documents

[AD 1]	TP3-J0-STB-116-CNES	Jason-2 Operations Services Specification
[AD 2]	TP3-J0-STB-44-CNES	OSTM/Jason-2 System Requirements
[AD 3]	TP3-J0-AQ-139-CNES	OSTM/Jason-2 4 Partner Mission Assurance Specs
[AD 4]	TP3-JS-IF-200-CNES	Jason-2 Ground System Interfaces
[AD 5]	TP3-JS-STB-110-CNES	OSTM/Jason2 Ground System Requirements, Architecture and Operations Concepts
[AD 6]	EUM/JAS/PLN/03/0003	EUMETSAT OSTM Jason 2 Management Plan
[AD 7]	WMO-No. 386	WMO Manual on the Global Telecommunications System
[AD 8]	SALP-ST-M-EA-15704-CN	SALP Products Specification – Volume 1: JASON-2 User Products, Version 2.3, 6/11/2008

1.3 Reference Documents

[RD 1]	EUM/JAS/REP/08/0013	Test plan and report for BUFR conversion software

2 SOFTWARE SPECIFICATION

2.1 Purpose

The software is required to take JASON2 OGDR data in netCDF (as per [AD 8]) and create from them files containing the equivalent BUFR encoded data. The files should structured and named ready for dissemination (as per [AD 7]).

2.2 Scope

The software is intended to be run operationally at EUMETSAT and at NOAA's site in Washington. It should also be possible to run the same software at CNES, and in a development environment at EUMETSAT. As the IGDR data are in the same netCDF format, they could also be reformatted using the same software. The software is also intended to run at the Australian Bureau of Meteorology (BoM) in order to convert IGDR data into BUFR for archiving purposes.

2.3 Constraints

The format conversion software is written in standard C. It has to be capable of running on different platforms (at EUMETSAT, CNES, NOAA and BoM). In order to support this portability constraint, two mitigating strategies have been adopted:

- The types `long int` and `unsigned long int` are each allocated 32 bits on some machines (such as dcomo01 and xlabs01) and 64 bits on others (such as jdo2ns01). In order to ensure compatibility, these types have been replaced by `int32_t` and `uint32_t` respectively. These types are always allocated 32 bits.
- On little-endian machines such as xlabs01, integers are written from the least significant to the most significant byte. On big-endian machines such as dcomo01 and jdo2ns01 the most significant byte is written first. As an example, 66051 would be stored as 0X00010203 on a little-endian machine and 0X03020100 on a big-endian machine. This issue is resolved by (i) using the `fwrite` to write data to files, as `fwrite` writes the data byte by byte using `putc` internally, and (ii) using the `htonl` function to ensure the bytes are always in the same order.

The software will be used as a data driven plug-in. As such, it takes one argument which is used to specify the input file.

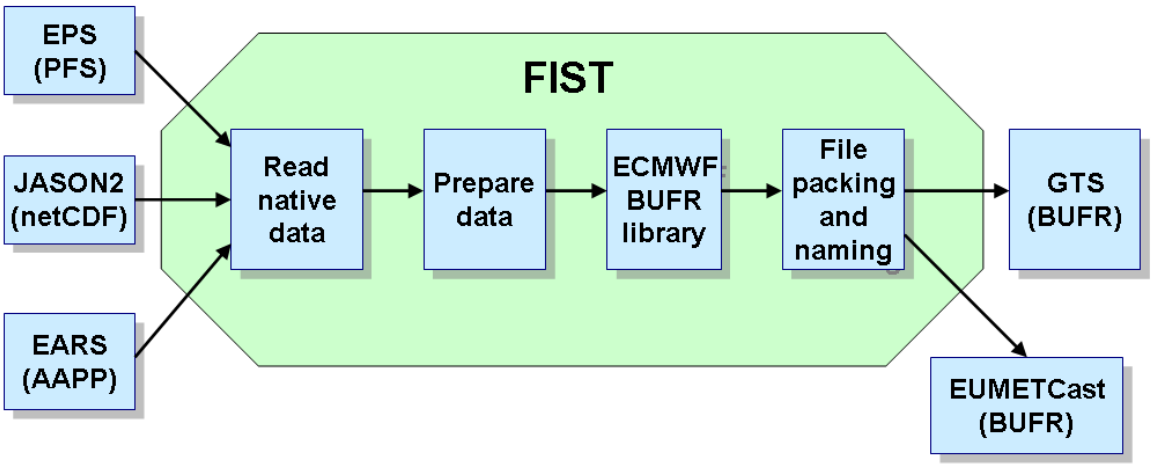
3 BUFR FORMATTING SOFTWARE DESIGN

The software used to reformat the JASON2 OGDR data from netCDF to BUFR is an instance of FIST (Format Interface Set of Tools). FIST is already used operationally at EUMETSAT for the reformatting of EARS data (from AAPP format to BUFR) and in the EPS ground segment for reformatting Metop and POES data (from PFS format to BUFR).

The data flow and general design of FIST is shown in the diagram below.

- Data are supplied in their native format, and read in according to the specification of that format. For the case of the JASON2 OGDR data, these will be in netCDF, structured as per [AD 8].
- The data are then prepared for encoding into BUFR. This involves steps such as changing physical units and/or setting of processing flags and code table entries. In principle, as much of the data processing as possible should be performed before the data are delivered to FIST. This keeps the data processing functions in one place, and maintains the role of FIST as a reformatting tool. For the case of JASON-2 OGDR data, the only checks applied to the data are to replace out of limit values and "netCDF fill values" by "missing" indicators in order to stop the encoding library reporting errors. Negative temperatures can be reported have been reported in the netCDF data as a result of overflow of signed 16 bit integers (AR 86); in this case the values are set to "missing" in BUFR and a log message is raised.
- The encoding into BUFR is carried out using the software library provided by ECMWF.
- Once the data have been encoded, they are packed into files ready for dissemination either globally via the GTS, or via EUMETSAT's DVB multicast service, EUMETCast. The files are packed and named in accordance with the WMO Manual on the GTS [AD 7].

The same files are used by EUMETSAT for both the GTS and EUMETCast dissemination. Identical files are also produced with a second file name in order to meet NOAA's specific requirements.



The encoded data will use the new BUFR Table D sequence descriptor 3-40-010. This sequence, together with the associated new BUFR Table B element descriptors and their code and flag tables, were designed specifically for the JASON2 OGDR data. They are in the process of being formally given operational status by WMO, and are anticipated to form part of Version 14 of BUFR Master Table 0 (Meteorology). The data will be encoded using BUFR Edition 4, and will make use of Version 14 of BUFR Master Table 0 until Version 15 is formally available.

4 BUFR FORMATTING SOFTWARE USER DOCUMENTATION

4.1 Introduction

This software forms part of FIST, a suite of similar programs used for converting various types of meteorological data from their native format to BUFR. This software acts as an interface to, and relies upon the freely available ECMWF BUFR encoding software library.

More information about JASON 2 can be found here: http://www.osd.noaa.gov/ostm/

More information about netCDF can be found here:
http://www.unidata.ucar.edu/software/netcdf/

More information about BUFR can be found here:
http://www.wmo.int/pages/prog/www/WMOCodes/Guides/BUFRCREXPreface_en.html

More information about the ECMWF BUFR encoding software can be found here:
http://www.ecmwf.int/products/data/software/bufr.html

4.2 Hardware requirements and compatibility

In order to proceed with the installation, the target host requires:

- A C compiler for FIST and the decoding utilities
- A FORTRAN compiler for building the ECMWF BUFR encoding library
- A standard installation of netCDF (development performed using version 3.6.2-1)

Hardware constraints such as required disk space, memory and processor speed are TBD.

4.3 ECMWF BUFR encoding library

The ECMWF BUFR encoding library should be downloaded from:
http://www.ecmwf.int/products/data/software/download/bufr.html

Versions 000320, 000340, 000350 and 000380 have been tested. It is recommended to use version 000380 as this has been successfully tested on several platforms including jd02ns01, and will be the most recent supported version from ECMWF for some time.

In order to build and install the library, the following steps should be performed:

1. If the downloaded file name ends "`.tar.tar`" rename it to end in "`.tar.gz`" (for example: `mv bufr_000380.tar.tar bufr_000380.tar.gz`)
2. Uncompress the .gz file (for example: `gunzip bufr_000380.tar.gz`)
3. Unpack the archive file (for example: `tar xvf bufr_000380.tar`)
4. Move to the location of the software (for example: `cd bufr_000380`)
5. Build and install the software (`build_library`).
    a. On a 32 bit machine (such as dcomo01) in answer to the installation questions, i) select [`n`] to work with default compilers, ii) select [`n`] to work with 32 bit reals, and iii) select [`.`] for the installation directory.
    b. On a 64 bit machine (such as jdo2ns01) in answer to the installation questions, i) select [`y`] to work with gcc and g77, ii) select [`y`] to work with 64 bit reals rather than the default 32 bit reals, and iii) select [`.`] for the installation directory.

This process should result in the creation of an archive library (`libbufr.a` on a 32 bit machine, and `libbufrR64.a` on a 64 bit machine) containing the BUFR encoding and decoding functions. If there are problems it may be necessary to adjust the make file. It is possible to consult ECMWF for advice (for example via eMail to software.services@ecmwf.int).

4.4 Environment variables

The following environment variables should be set:

BUFR_TABLES this should point to a directory containing the BUFR tables. See Section 5.
For example: `setenv BUFR_TABLES /home/FIST/BUFR_Tables/`

LD_LIBRARY_PATH this should include to the location of the netCDF library installation.
For example: `setenv LD_LIBRARY_PATH $LD_LIBRARY_PATH:/opt/netcdf-3.6.2-1/lib`

4.5 BUFR tables

The BUFR tables required by the ECMWF software should be put into the directory pointed to by the environment variable BUFR_TABLES (as per Section 4). As of February 2010, the BUFR tables are called:
B0000000000000014000.TXT
C0000000000000014000.TXT
D0000000000000014000.TXT

4.6 Encoding software

The following source code files are required:
`J2_OGDR_to_BUFR.c`
`J2_OGDR_to_BUFR.h`

These should be compiled using the make file `J2_OGDR_to_BUFR.mk` to generate the executable program `J2_OGDR_to_BUFR`

The make file may need to be adjusted to refer to the appropriate compilers and/or paths to the netCDF and BUFR libraries. For example, the linking/loading line in the make file should end with `-lbufr` on 32 bit machines and `-lbufrR64` on 64 bit machines.

5 PROGRAM EXECUTION

5.1 Reformatting data

The software looks for a file specified by its first argument. This argument should either be the netCDF OGDR file to be reformatted, or a symbolic link to that file. For example, if the file to be reformatted is called JA2_OPN_2PAP999_999_20060628_020002_20060628_035912 and is in a file called /home/FIST/data_in then the following steps would be performed:

- Make a link to the input file (`ln -s /home/FIST/data_in/ JA2_OPN_2PAP999_999_20060628_020002_20060628_035912 .`)
- Run the reformatting software (`J2_OGDR_to_BUFR JA2_OPN_2PAP999_999_20060628_020002_20060628_035912`)

In these circumstances, the program should generate two identical output files in the local directory called:
`W_XX-EUMETSAT-Darmstadt,SURFACE+SATELLITE,JASON2+OGDR_C_EUMS_20071112172350_A_999_999_20060628035912.bin`
And
`JA2_OPB_2PAP999_999_20060628_020002_20060628_035912`

These files will contain the JASON 2 OGDR data in BUFR. The files will be named and structured in accordance with the relevant ICD (TP3-JS-IF-200-CNES, [AD 4]) and the WMO Manual on the Global Telecommunication System, [AD 7].

5.2 Command line options

The software can also be supplied with one of four specific command line arguments in order to display certain pertinent information. These options are used as follows:

5.2.1 WMO file name

`J2_OGDR_to_BUFR --wmo-file-name [input file name]`
The file name which would be used for the GTS is output, but no files are written.
Entering: `J2_OGDR_to_BUFR --wmo-file-name JA2_OPN_2PAP999_999_20060628_020002_20060628_035912`

produces the output:
```
W_XX-EUMETSAT-
Darmstadt,SURFACE+SATELLITE,JASON2+OGDR_C_EUMS_20071112172350_
A_999_999_20060628035912.bin
```

5.2.2 FIST software version

`J2_OGDR_to_BUFR -- fist-version`

The version of the BUFR formatting software (FIST) is output, and no files are written.
Entering: `J2_OGDR_to_BUFR -- fist-version`
produces the output:
```
  1.8
```

5.2.3 BUFR table versions

`J2_OGDR_to_BUFR -- bufr-table-versions`

The version of the BUFR tables is output, but no files are written.
Entering: `J2_OGDR_to_BUFR -- bufr-table-versions`
produces the output:
```
 14.0
```

5.2.4 ECMWF software library version

`J2_OGDR_to_BUFR -- ecmwf-library-version`

The version of the ECMWF BUFR encoding software library is output, but no files are written.
Entering: `J2_OGDR_to_BUFR -- ecmwf-library-version`
produces the output:
```
  380
```

6 DECODING THE BUFR OGDR DATA

The result of the BUFR encoding process will be one file containing many BUFR messages. This file can be decoded using any generic BUFR decoding software which has access to the relevant tables (Tables B and D and Code and Flag Tables relating to Master Table 0, Master Table version 14, Local Table version 0, duly augmented by the agreed descriptors for JASON2 OGDR data).

As an example the file can be split into individual BUFR parts by using the program “buffy”. This can be compiled from the source code very simply (`cc –o buffy buffy.c`). It runs taking one argument, the name of the file to be processed, and generates one file for each BUFR part of the input file as output. If my_file.buf conatins 3 BUFR messages, then entering:
```
buffy my_file.buf
```
would produce 3 output files; BUFR0001, BUFR0002 and BUFR0003, each containing a complete set of BUFR data.

The files containing a single BUFR message can be analysed using a program such as DecJASON2_OGDR. This can be compiled from the source code very simply (`cc –o DecJASON2_OGDR DecJASON2_OGDR.c -lm`). It runs taking one argument, the name of the file to be processed, and writes decoded information to the standard output.

APPENDIX A ACRONYMS

BoM	Australian Bureau of Meteorology
BUFR	Binary Universal Format for the Representation of Meteorological Data
CNES	Centre National d'Etudes Spatiales
ECMWF	European Centre for Medium-range Weather Forecasts
EUMETSAT	European Organisation for the Exploitation of Meteorological Satellites
EPS	EUMETSAT Polar System
FIST	Format Interface Set of Tools
GTS	Global Telecommunications System
ICD	Interface Control Document
IGDR	Interim Geophysical Data Record (note: this denomination is replaced by NRT products throughout the document)
NOAA	National Oceanographic and Atmospheric Administration
netCDF	network Common Data Format
OGDR	Operational Geophysical Data Record (note: this denomination is replaced by NRT products throughout the document)
PFS	Product Format Specification
POES	Polar Operational Environment Satellite
TBD	To be defined
WMO	World Meteorological Organisation

APPENDIX B TBDS

Hardware constraints such as required disk space, memory and processor speed remain unspecified.

APPENDIX C MAPPING OF QUALITY FLAGS

This appendix summarises the mapping of the "flag" information inside the netCDF data to the various BUFR code and flag tables which are used.

netCDF variable name	BUFR Table B descriptor	Comments
surface_type	0-08-029	Interpretation as for netCDF
alt_echo_type	0-08-074	Interpretation as for netCDF
rad_surf_type	0-08-077	netCDF: 0 = ocean, 1 = land BUFR: 0 = land, 1 = sea
interp_flag_mss	0-40-011	Bit 1, interpretation as for netCDF
interp_flag_ocean_tide_sol1	0-40-011	Bit 2, interpretation as for netCDF
interp_flag_ocean_tide_sol2	0-40-011	Bit 3, interpretation as for netCDF
interp_flag_meteo	0-40-011	Bit 4, interpretation as for netCDF
orb_state_flag_diode	0-25-097	Interpretation as for netCDF
alt_state_flag_oper	0-25-095	Interpretation as for netCDF
qual_alt_1hz_range_ku	0-25-098	Bit 1, interpretation as for netCDF
qual_alt_1hz_range_c	0-25-098	Bit 2, interpretation as for netCDF
qual_alt_1hz_swh_ku	0-25-098	Bit 3, interpretation as for netCDF
qual_alt_1hz_swh_c	0-25-098	Bit 4, interpretation as for netCDF
qual_alt_1hz_sig0_ku	0-25-098	Bit 5, interpretation as for netCDF
qual_alt_1hz_sig0_c	0-25-098	Bit 6, interpretation as for netCDF
qual_alt_1hz_off_nadir_angle_wf_ku	0-25-098	Bit 7, interpretation as for netCDF
qual_alt_1hz_off_nadir_angle_pf	0-25-098	Bit 8, interpretation as for netCDF
qual_inst_corr_1hz_range_ku	0-25-099	Bit 1, interpretation as for netCDF
qual_inst_corr_1hz_range_c	0-25-099	Bit 2, interpretation as for netCDF
qual_inst_corr_1hz_swh_ku	0-25-099	Bit 3, interpretation as for netCDF
qual_inst_corr_1hz_swh_c	0-25-099	Bit 4, interpretation as for netCDF
qual_inst_corr_1hz_sig0_ku	0-25-099	Bit 5, interpretation as for netCDF
qual_inst_corr_1hz_sig0_c	0-25-099	Bit 6, interpretation as for netCDF
rain_flag	0-21-144	Interpretation as for netCDF
rad_state_flag_oper	0-25-096	Bit 1, interpretation as for netCDF
qual_rad_1hz_tb187	0-40-012	Bit 1, interpretation as for netCDF
qual_rad_1hz_tb238	0-40-012	Bit 2, interpretation as for netCDF
qual_rad_1hz_tb340	0-40-012	Bit 3, interpretation as for netCDF
interp_flag_tb	0-40-013	Interpretation as for netCDF
ice_flag	0-21-169	Interpretation as for netCDF
alt_state_flag_c_band	0-40-023	Bit 1, interpretation as for netCDF
alt_state_flag_c_band_status	0-40-023	Bit 2, interpretation as for netCDF
alt_state_flag_ku_band_status	0-40-023	Bit 3, interpretation as for netCDF
alt_state_flag_band_seq	0-40-023	Bit 4, interpretation as for netCDF
ecmwf_meteo_map_avail	0-40-024	Interpretation as for netCDF
interp_flag_mdt	0-40-025	Interpretation as for netCDF