FM94 BUFR

Decoding Software for ERS Data

Detailed Design Document

(Software Release V5.01)

ER-SD-UKM-GS-0002
Version 4, Issue 2
13 September 1999

FM94 BUFR Decoding Software for ERS Data
Detailed Design Document
(Software Release V5.01)

Contents

Part 2 - Component Design Specifications

1. **ASC2EB**
2. **BUFRER**
3. **BVALUE**
4. **CERSORB_BLOCK**
5. **CHGCASE**
6. **CODFLG**
7. **DAY_OF_YEAR**
8. **DEBUFR**
9. **DEBUFR_FDP**
10. **DEBUFR_UAT**
11. **DEBUFR_URA**
12. **DEBUFR_UWA**
13. **DEBUFR_UWI**
14. **DECOD4**
15. **DECODE_FDP**
16. **DESFXY**
17. **DT2I**
18. **DT_NOW**
19. **EB2ASC**
20. **ERRCODES**
21. **ERSORB**
22. **FDPTYPES**
23. **FRACTION_OF_DAY**
24. **FXYDES**
25. **GETARG**
26. **GETDIR**
27. **GETENV**
28. **GTS_IO**
29. **HDR_CF**
30. **IARGC**
31. **IPBTAB**
32. **JULIAN_DATE**
33. **JULIAN_OF_YEAR**
34. **JULIAN_TIME**
35. **JULIAN_TO_DATIM**
36. **OPEN_FDP**
37. **ORBPAR**
38. **STATIONS**
39. **STRLEN**
40. **TABLEA**
41. **TABLEB**
42. **TABLED**
43. **UAT_CF**

44. **UAT_IO**
45. **URA_CF**
46. **URA_IO**
47. **UWAFIXED**
48. **UWA_CF**
49. **UWA_IO**
50. **UWI_CF**
51. **UWI_IO**

FM94 BUFR Decoding Software for ERS Data
Detailed Design Document
(Software Release V5)

Part 1 - General Description

1. Introduction

1.1 Purpose

To decode ERS Low Bit Rate (LBR) Fast Delivery (FD) products from the World Meteorological Organisation (WMO) standard FM94 BUFR format required for transmission to users via the WMO Global Telecommunications System (GTS).

1.2 Scope

Four types of FD products are handled:-

UAT - ATSR (ASST) data,
URA - Radar Altimeter data,
UWA - Wave Scatterometer data,
UWI - Wind Scatterometer data.

The decoding software is written in FORTRAN 77 for OpenVMS V6.0 or higher on DEC VAX and Alpha platforms. Where possible operating system specific features, such as command line calls and message logging, are contained in subroutines so that they may be replaced for other environments.

1.3 Definitions, acronyms and abbreviations

ASST	Averaged Sea Surface Temperature
ATSR	Along Track Scanning Radiometer
BUFR	Binary Universal Format for data Representation, a WMO approved encoding method otherwise known as Format Number 94 (FM94)
DEC *	Digital Equipment Corporation
DCL	Digital Command Language (VMS command line interpreter)
ECMWF	European Centre for Medium-range Weather Forecasts
ERS	European Remote Sensing satellite
ESA	European Space Agency
FDP	Fast Delivery Product
GTS	Global Telecommunications System
ISS	Interface Sub-System (ESA facility for the distribution of ERS data)
LBR	Low Bit Rate
MPH	Main Product Header
PCD	Product Confidence Data
RAL	Rutherton Appleton Laboratory
RMS	Record Management System
SADIST	Synthesis of ATSR Data Into Sea-surface Temperatures. Processing software system for ATSR data (RAL). SADIST (V600) for ATSR on ERS-1 only, upgraded to SADIST-2 for ATSR-2 on ERS-2

SPH	Specific Product Header
SST	Sea Surface Temperature
VMS *	Virtual Memory System (DEC VAX operating system)
WMO	World Meteorological Organisation

* DEC, VAX, VMS and OpenVMS are registered trademarks of the Digital Equipment Corporation.

1.4 References

[1]. ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.

[2]. Manual on Codes. WMO, GENEVA, 1995. No.306, Vol.I-2, Part B, Binary Codes (with Supplement No.3, August 1998).

[3]. Manual on the Global Telecommunications System. Vol.1, Part II. WMO, Geneva, 1992.

[4]. ESA software engineering standards, ESA PSS-05-0 Issue 2,, February 1991.

[5]. P.Bailey. SADIST-2 V100 Products. RAL, ER-TN-RAL-AT-2164, 6 September 1995.

[6]. DEC FORTRAN User Manual for OpenVMS systems. Part No: AA-PUYPA-TE.

[7]. ESA/ESRIN. FM94 BUFR Decoding software for ERS data., Software User Manual. ER-MA-UKM-GS-0002, Version 4.

1.5 Overview

Since as far as ESA/Earthnet is concerned, decoding is only done for testing purposes, it does not need to interface to ISS. The distributed version therefore does not use formal message logging files and formats, and messages are output to SYS$OUTPUT. The program can be invoked via a DCL foreign command or called from a user program by linking in the BUFR object library. See Ref.7 for installation and usage details.

To invoke from a DCL foreign command:-

$ DECODE :== $disk:[dir]DECODFDP.EXE

$ DECODE [gtsfile] [uatfile] [urafile] [uwafile] [uwifile] [dmpfile]

where:

disk:[dir] is the disk and directory containing the executable code.

To call from a user program:-

CALL DEBUFR_FDP(**gtsfile, uatfile, urafile, uwafile, uwifile, dmpfile**)

In both cases:-

gtsfile is the VAX file specification for the encoded BUFR GTS input file.

uatfile is the full VAX file specification of the UAT output FDP file in ESA format.

urafile is the full VAX file specification of the URA output FDP file in ESA format.

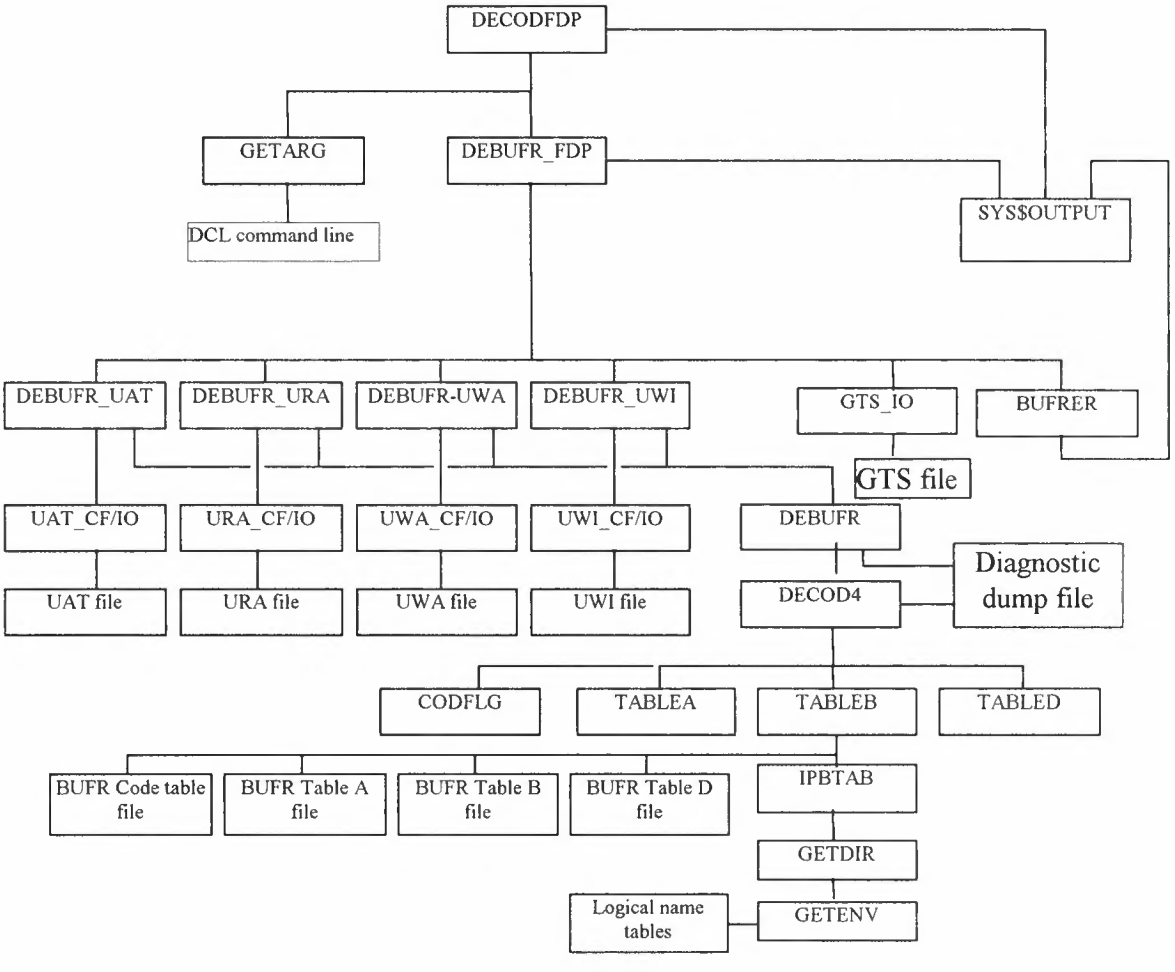
uwafile is the full VAX file specification of the UWA output FDP file in ESA format.

uwifile is the full VAX file specification of the UWI output FDP file in ESA format.

dmpfile is an optional decode diagnostic dump file

A diagrammatic overview of the decoding process is shown in Figure 1. The scheme assumes that each product type is to be output to a separate, newly-created file, with one or more products per file in the same format as read into the encoding program. The input file should contain one or more complete GTS bulletins of no more than 15,000 bytes per record which itself contains one (and only one) BUFR message. This file may contain one or more product types, which may be mixed in any order.

Figure 1 **Overview of the ERS-1/ERS-2 product BUFR decoding scheme**



On activation, the following steps occur:

1. If invoked via a DCL command line the main level program, DECODE_FDP, reads the DCL command line arguments (subroutine GETARG) and calls the encoding routine DEBUFR_FDP.

2. The DEBUFR_FDP routine first opens the input file; if it cannot be opened, the routine returns to the main program.

3. As a GTS message is read in, it is scanned for the ETX terminator (normally the last byte read), the message sequence number and product type from the GTS abbreviated header. The appropriate routine is then called to decode that message. Unknown product types are discarded.

4. On the first invocation of the BUFR kernel decoder the BUFR tables are located and loaded. This is achieved by translating the DCL logical name 'BUFRDIR' which must point to the directory containing the binary table files.

5. The individual DEBUFR_ppp routines (where ppp is UWA, UWI, URA or UAT) decode the message into the plain BUFR parameters, then post-process back to ESA format and output the product to its own file via the ppp_CF and ppp_IO routines. Any BUFR or I/O errors are signalled back to the calling routine. New individual product files are only created and opened when the first product of that type has been successfully decoded and is ready for output. If any individual output file cannot be opened, or an output error occurs writing to it during decoding, that product type is disabled from further processing but other products will continue normally.

6. DEBUFR_FDP displays any error or status messages and loops around for more GTS messages until the end of the GTS file is detected, when all files are closed and control returns to the main program or user calling routine.

In normal operation the file specification 'dmpfile' should either be blank or not specified on the DCL command line, or be blank in the subroutine interface call. In each case no diagnostic output is generated. If a valid file name is given the decoder will write a diagnostic dump of the BUFR headers and data in plain text format, interpreting parameter names, units, values and code table meanings etc. ***The user should note that this file can be very large; the option is intended for trouble shooting only and not for normal output.***

2. Project Standards, Conventions and Procedures

2.1 Design standards

The code is highly modular, with each routine undertaking a specific and limited task, and in a top-down manner.

2.2 Documentation standards

To ESA Software Engineering Standards, (see Ref [4]).

2.3 Naming conventions (including files, programs, modules, etc.)

2.3.1 File names.

Definition of all file names is optional. If they are not defined the GTS input file name defaults to FDP_BUFR.GTS and the output files are created with names of the form 'Uxx_ddhhmm.Ent' where 'Uxx' is the product type, 'ddhhmm' is the day, hour and minute of the first of that type of product in the GTS file (taken from the WMO header), 'n' is 1 for ERS-1 data and 2 for ERS-2 data and 't' is blank for operational data or 'T' for test data. BUFR messages for ERS-1 and ERS-2, or operational and test modes, should not be mixed in a single GTS file if this default naming is being used to identify these attributes. If data for a particular product type is not in the GTS message then the corresponding FDP product file will not be created. The default for the diagnostic dump file is not to generate dump output. If the disk and directory information is not defined the current directory is assumed.

2.3.2 Module names.

To aid transfer to non-DEC platforms the names of source code files are restricted to '8.3' format with no underscores.

2.4 Programming standards

As far as possible all code is written in ISO-standard FORTRAN-77. Only common extensions to the standard FORTRAN 77 such as the use of IMPLICIT NONE, to enforce the declaration of all variables, INTEGER*4 declarations and variable names of length greater than 6 characters are used except in clearly flagged areas where it has been necessary to use DEC-FORTRAN extensions to meet requirements for user and file system interfaces. Software components in which DEC-FORTRAN extensions have been used are marked VAXFORT in the TYPE area in the Component Design Specifications section in Part 2 of this document; components using only FORTRAN-77 are marked FORT77.

2.5 Software development tools

DEC FORTRAN 77 compiler - see Ref [6] - and EVE/EDT text editor.

2.6 Source code listings

The source code listings are available in the BUFR_ROOT:[.SOURCE] sub-directory that is created during the installation procedure.

Part 2 - Component Design Specifications

1.

IDENTIFIER	ASC2EB			
TYPE	SUBROUTINE (B0016 in BUFRKERN.FOR) - (FORT77)			
PURPOSE	Convert ASCII characters to EBCDIC			
ARGUMENTS	NC	(Entry)	I4	Number of characters to convert
	STRING	(Entry/Exit)	C*	Character string
FUNCTION	Converts first 'NC' characters of 'STRING' from ASCII to EBCDIC representation.			
CALLED FROM	DEBUFR_FDP, DECOD4			

2.

IDENTIFIER BUFRER

TYPE SUBROUTINE (B0151 in BUFRKERN.FOR) - (FORT77)

PURPOSE Return BUFR error message from error code

ARGUMENTS

IERR	(Entry)	I4	BUFR error code (000-519)
ERRMSG	(Exit)	C*	Error message
ERRUNIT	(Entry)	I4	Error message output stream (suppressed if <=0)

COMMON /BUFERR/

IEL	(Read)	I4	Index of element in error
IOB	(Read)	I4	Index of observation in error
IDES	(Read)	I4	Descriptor value in error
IVAL	(Read)	I4	Observation value in error (indicates encode or decode for x01 errors)

FUNCTION Returns an error message corresponding to a BUFR error code. Codes are of the form "SCE" where S is the BUFR section number (0-5), C=0 for encoding, 1 for decoding and E is the specific error. (exception: x01 errors which can potentially occur on encode and decode.) Error value zero means all OK, so this one returns 'OK' message. Unknown codes are returned 'unknown error'. The returned message may be up to 25 characters long. If 'ERRUNIT' is >0, the message is output to that stream unit number formatted into multiple lines, depending on the error; if <=0, no messages are written.

SUBPROGRAMS DESFXY, TABLEB

INTRINSICS INDEX, MOD

CALLED FROM DEBUFR_FDP, ERROUT

3.

IDENTIFIER BVALUE

TYPE FUNCTION (B0007 in BUFRKERN.FOR) - (FORT77)

PURPOSE Get value from bits in BUFR string

ARGUMENTS

MESAGE	(Entry)	C*	BUFR bit string
IBEFOR	(Entry/exit)	I4	Number of bits to skip
WIDTH	(Entry)	I4	Field width in bits
BVALUE	(Function)	I4	Extracted value

FUNCTION Extracts a value from a BUFR-encoded bit string, 'WIDTH' bits wide, first skipping 'IBEFOR' bits from the beginning of the string. 'IBEFOR' is updated to the end of the field before return. Field width may span one or more byte boundaries.

INTRINSICS ICHAR, MOD

CALLED FROM DEBUFR, DECOD4, ENBUFR

4.

IDENTIFIER	CERSORB_BLOCK
TYPE	BLOCK DATA (G0402 in ERSORB.FOR) - (FORT77)
PURPOSE	Dummy BLOCK DATA to replace ESA ERSORB version
FUNCTION	This named BLOCK DATA module provides a dummy version to go with the dummy ERSORB subroutine (ref G0401) in order to make linking transparent as to whether the real or dummy ERSORB is present in the object library.
INCLUDED IN	ERSORB
REFERENCES	B. Duesmann and H Klinkrad: "ERS Orbit Propagator Software (ERSORB). Interface and Installation Guide". ESA Document ER-TN-ESA-GS-0003 Issue 3.2, May 1994.

5.

IDENTIFIER	CHCASE			
TYPE	FUNCTION (G0104 in UTILS.FOR) - (FORT77)			
PURPOSE	Changes any string to lower or upper case			
ARGUMENTS	CASE	(Entry)	C*	'L' or 'l' for lower case or 'U' or 'u' for upper case.
	STRING	(Entry/exit)	C*	Character string for conversion.

FUNCTION An arbitary ASCII character string is converted to all lower case or all upper case, depending on the first character of CASE. Non-alphabetic characters (numerals, punctuation, etc.) are unaffected. If CASE is not one of 'L', 'l', 'U' or 'u', then the string is not altered.

INTRINSICS STRLEN

CALLED FROM DEBUFR_FDP, DECODE_FDP

6.

IDENTIFIER CODFLG

TYPE SUBROUTINE (B0012 in BUFRKERN.FOR) - (FORT77)

PURPOSE To look up the description of a code figure or flag

ARGUMENTS

X	(Entry)	I4	X part of descriptor
Y	(Entry)	I4	Y part of descriptor
VALUE	(Entry)	I4	Code or flag value
WORDS	(Exit)	C*	Text description of entry

COMMON / BUFTAB / LOADED
/ BUFRCC / CFTEXT
/ BUFRCI / NCODES, INDEX

FUNCTION The data read consists of:
- The number of tables (NCODES)
- An index entry for each table consisting of the descriptor with a pointer to the count below
- For each table the number of code figures defined (1 octet) followed by the descriptions each preceded by a 1-octet length.
Return with word blank if descriptor or code figure not found or if value missing.
BUFR tables are automatically loaded if not already in memory.

SUBPROGRAMS IPBTAB

INTRINSICS ICHAR, MOD

CALLED FROM DEBUFR, DECOD4

7.

IDENTIFIER	DAY_OF_YEAR			
TYPE	FUNCTION (G0206 in DATETIME.FOR) - (FORT77)			
PURPOSE	Convert calendar date to day of year			
ARGUMENTS	YR	(Entry)	I4	Year
	MO	(Entry)	I4	Month of year (1-12)
	DY	(Entry)	I4	Day of month (1-31)
	DAY_OF_YEAR	(Function)	I4	Day of year (1-366)
FUNCTION	Converts between calendar date (day,month,year) and n-th day into year (1-365 or 366 for leap years). The direction of the conversion is determined by the value of 'INV'			
INTRINSICS	MAX, MIN, MOD			
CALLED FROM	JULIAN_TO_DATIM			

8.

IDENTIFIER DEBUFR

TYPE SUBROUTINE (B0003 in BUFRKERN.FOR) - (FORT77)

PURPOSE Decode a BUFR message (data only) (Editions 1 - 3)

ARGUMENTS

MESAGE	(Entry)	C*	BUFR bit string
DUNIT	(Entry)	I4	Unit number for display (<=0 to suppress display)
DATIM	(Exit)	I4	5-element date/time of bulletin (YEAR, MONTH, DAY, HOUR, MINUTE)
DATYP	(Exit)	I4	2-element data type and sub-type
SECTN2	(Exit)	L4	TRUE (.T.) if section 2 present
DESCR	(Exit)	I4	Expanded list of descriptors
NDESC	(Exit)	I4	Number of descriptors
VALUES	(Exit)	I4	NOBS*NDESC data values (observations)
NOBS	(Exit)	I4	Number of observations
NAMES	(Exit)	C*	Decoded character data
CMPRES	(Exit)	L4	TRUE (.T.) if data was compressed
IERR	(Exit)	I4	Error code (0=OK, else error)

COMMON / BUFERR / IEL, IOB, IDES, IVAL - parameters on error

IEL	I4	Element at error
IOB	I4	Observation at error
IDES	I4	Descriptor at error
IVAL	I4	Value/increment etc at error

FUNCTION Decodes a BUFR message (Editions 1, 2 and 3); checks all sections but skips any data in section 2 (just flags whether present or not). If compression was in effect, all observations for one element are followed by all observations for the next element, etc., Otherwise all elements are repeated for each observation. Error return codes:

x01 - exceeded length of BUFR string in section x
011 - section 0 'BUFR' not found
311 - no observations
312 - no descriptors
411 - no entry for descriptor in TABLE B or TABLE D
412 - operator descriptor (F=2), but no operation defined (X>5)
413 - delayed replication (Y=0), but no count in data
414 - delayed replication with compression, but counts vary
415 - new reference value(s), but end of definition not marked
416 - new reference values with compression, but values vary
417 - all values missing with compression, but increment width nonzero
418 - data repetition with more than one descriptor
419 - data repetition but compression

511 - message does not end with '7777'

SUBPROGRAMS BVALUE, CODFLAG, DECOD4, DESFXY, EB2ASC, STRLEN, TABLEA

INTRINSICS BTEST, ICHAR, INDEX, LEN

CALLED FROM DEBUFR_UWA, DEBUFR_UWI, DEBUFR_URA, DEBUFR_UAT

9.

IDENTIFIER	DEBUFR_FDP			
TYPE	SUBROUTINE (B2011 in DEBUFFDP.FOR) - (FORT77)			
PURPOSE	Decode and output ERS-1/2 fast delivery products			
ARGUMENTS	GTSDSN	(Entry/exit)	C*	Name of GTS input message file (default: 'FDB_BUFR.GTS')
	UATDSN	(Entry/exit)	C*	Name of UAT output FDP file (default: 'UAT_ddhhmm.Ent')
	URADSN	(Entry/exit)	C*	Name of URA output FDP file (default: 'URA_ddhhmm.Ent')
	UWADSN	(Entry/exit)	C*	Name of UWA output FDP file (default: 'UWA_ddhhmm.Ent')
	UWIDSN	(Entry/exit)	C*	Name of UWI output FDP file (default: 'UWI_ddhhmm.Ent')
	DMPDSN	(Entry/exit)	C*	Name for decoded dumps (default: suppress dump)
	IERR	(Exit)	I4	Exit error/status code 0 = OK 100-599 = BUFR coding error 700-799 = FDP product error 898 = GTS file open error 899 = GTS file read error 998 = FDP open error 999 = FDP write error

COMMON	/BUFERR/			
	IEL	(Read)	I4	Index of element in error
	IOB	(Read)	I4	Index of observation in error
	IDES	(Read)	I4	Descriptor value in error
	IVAL	(Write)	I4	Observation value in error (indicates encode or decode for x01 errors or Main Product Header product type read for 702 errors or FORTRAN error code for Input/Output errors)

INCLUDES	ERRCODES.FOR	Error code definitions
	FDPTYPES.FOR	Fast Delivery Products identifier definitions
	SYSTEM.FOR	System definitions

FUNCTION Inputs ERS FDP data (mixed products) in GTS/BUFR format and decodes them. Output in FDP formats. If the GTS file name is blank, 'FDP_BUFR.GTS' is used. If FDP file names are blank, files are created with names of the form **Uxx__ddhhmm.Ent** where **Uxx** is the product type, **ddhhmm** is the day, hour and minute of the first of that

type of product in the GTS file (taken from the WMO header), **n** is 1 or 2 for ERS-1 or ERS-2 and **t** is blank for operational data or T for test data. Note that if no data for any particular product is present in the GTS stream the corresponding FDP file will not be created. Error and information messages are output to file STD.OUT. If dump file name is not blank, diagnostic decoded data is dumped to it. Default is not to output dump data.

SUBPROGRAMS DEBUFR_UAT, DEBUFR_URA, DEBUFR_UWA, DEBUFR_UWI, ASC2EB, BUFRER, CHCASE, EB2ASC, GTS_IO, OPEN_FDP

INTRINSICS CHAR, INDEX, LEN

CALLED FROM DECODE_FDP

FILES	UNIT(*)	FILE(SYS$OUTPUT)	Messages	(Output)
	UNIT(11)	FILE(GTSDSN)	FDP GTS/BUFR data	(Input)
	UNIT(21)	FILE(UATDSN)	ATSR Fast Delivery data	(Output)
	UNIT(22)	FILE(URADSN)	Radar Altitude Fast Delivery data	(Output)
	UNIT(23)	FILE(UWADSN)	Wave Fast Delivery data	(Output)
	UNIT(24)	FILE(UWIDSN)	Scatterometer Fast Delivery data	(Output)
	UNIT(31)	FILE(DMPDSN)	Decode dump	(Output)

REFERENCES

1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 1994.
2) UWA processing algorithm specification, Version 2.0, ER-TN-ESA-GS-0342, ISSUE 1.0, Nov 1995
3) P.Bailey. SADIST-2 V100 Products. RAL, ER-TN-RAL-AT-2164, 6 September 1995..
4) Manual on codes, WMO-306, Vol. 1-2, Part B, 1999 (with supplement No.3, August 1998)
5) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

10.

IDENTIFIER DEBUFR_UAT

TYPE SUBROUTINE (B2012 in DEBUFFDP.FOR) - (FORT77)

PURPOSE Decode one UAT product from GTS/BUFR format.

ARGUMENTS

UATUNIT	(Entry)	I4	Unit number of UAT output file
DMPUNIT	(Entry)	I4	Dump output unit (0=suppress)
GTSMSG	(Entry)	C*	GTS message string
DTP	(Exit)	C24	Date/time of decoded product
IERR	(Exit)	I4	Error code 0 = UAT decoded and saved OK 100-599 = BUFR error 999 = file write error

INCLUDES

ERRCODES.FOR	Error code definitions
FDPTYPES.FOR	Fast Delivery Products identifier definitions

FUNCTION Decodes a single ATSR UAT product in BUFR format message, and writes it to file. The UAT output file must have been opened before the first call to this routine.

SUBPROGRAMS DEBUFR, UAT_CF, UAT_IO

CALLED FROM DEBUFR_FDP

FILES

UNIT(UATUNIT)	ATSR Fast Delivery data	(Output)
UNIT(DMPUNIT)	Decoded dump	(Output)

REFERENCES

1) P.Bailey. SADIST-2 V100 Products. RAL, ER-TN-RAL-AT-2164, 6 September 1995.
2) Manual on codes, WMO-306, Vol. 1-2, Part B, 1999 (with supplement No.3, August 1998)
3) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

11.

IDENTIFIER DEBUFR_URA

TYPE SUBROUTINE (B2013 in DEBUFFDP.FOR) - (FORT77)

PURPOSE Decode one URA product from GTS/BUFR format.

ARGUMENTS

URAUNIT	(Entry)	I4	Unit number of URA output file
DMPUNIT	(Entry)	I4	Dump output unit (0=suppress)
GTSMSG	(Entry)	C*	GTS message string
DTP	(Exit)	C24	Date/time of decoded product
IERR	(Exit)	I4	Error code 0 = URA decoded and saved OK 100-599 = BUFR error 999 = file write error

INCLUDES

ERRCODES.FOR	Error code definitions
FDPTYPES.FOR	Fast Delivery Products identifier definitions

FUNCTION Decodes a single radar altimeter URA product in BUFR format message, and writes it to file. The URA output file must have been opened before the first call to this routine.

SUBPROGRAMS DEBUFR, URA_CF, URA_IO

CALLED FROM DEBUFR_FDP

FILES

UNIT(URAUNIT)	Radar Altimeter Fast Delivery data	(Output)
UNIT(DMPUNIT)	Decoded dump	(Output)

REFERENCES

1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.
2) Manual on codes, WMO-306, Vol. 1-2, Part B, 1999 (with supplement No.3, August 1998)
3) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

12.

IDENTIFIER	DEBUFR_UWA			
TYPE	SUBROUTINE (B2014 in DEBUFFDP.FOR) - (FORT77)			
PURPOSE	Decode one UWA product from GTS/BUFR format.			
ARGUMENTS	UWAUNIT	(Entry)	I4	Unit number of UWA output file
	DMPUNIT	(Entry)	I4	Dump output unit (0=suppress)
	GTSMSG	(Entry)	C*	GTS message string
	DTP	(Exit)	C24	Date/time of decoded product
	IERR	(Exit)	I4	Error code 0 = UWA decoded and saved OK 100-599 = BUFR error 999 = file write error
INCLUDES	ERRCODES.FOR			Error code definitions
	FDPTYPES.FOR			Fast Delivery Products identifier definitions
FUNCTION	Decodes a single wave scatterometer UWA product in BUFR format message, and writes it to file. The UWA output file must have been opened before the first call to this routine.			
SUBPROGRAMS	DEBUFR, UWA_CF, UWA_IO			
CALLED FROM	DEBUFR_FDP			

FILES	UNIT(UWAUNIT)	Wave Scatterometer Fast Delivery data	(Output)
	UNIT(DMPUNIT)	Decoded dump	(Output)

REFERENCES

1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.
2) UWA processing algorithm specification, Version 2.0, ER-TN-ESA-GS-0342, ISSUE 1.0, Nov 1995
3) Manual on codes, WMO-306, Vol. 1-2, Part B, 1999 (with supplement No.3, August 1998)
4) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

13.

IDENTIFIER	DEBUFR_UWI			
TYPE	SUBROUTINE (B2015 in DEBUFFDP.FOR) - (FORT77)			
PURPOSE	Decode one UWI product from GTS/BUFR format.			
ARGUMENTS	UWIUNIT	(Entry)	I4	Unit number of UWI output file
	DMPUNIT	(Entry)	I4	Dump output unit (0=suppress)
	GTSMSG	(Entry)	C*	GTS message string
	DTP	(Exit)	C24	Date/time of decoded product
	IERR	(Exit)	I4	Error code 0 = UWI decoded and saved OK 100-599 = BUFR error 999 = file write error

INCLUDES	ERRCODES.FOR	Error code definitions
	FDPTYPES.FOR	Fast Delivery Products identifier definitions

FUNCTION Decodes a single scatterometer UWI product in BUFR format message, and writes it to file. The UWI output file must have been opened before the first call to this routine.

SUBPROGRAMS DEBUFR, UWI_CF, UWI_IO

CALLED FROM DEBUFR_FDP

FILES	UNIT(UWIUNIT)	Scatterometer Intermediate Product data	(Output)
	UNIT(DMPUNIT)	Decoded dump	(Output)

REFERENCES 1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.
2) Manual on codes, WMO-306, Vol. 1-2, Part B, 1999 (with supplement No.3, August 1998)
3) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

14.

IDENTIFIER DECOD4

TYPE SUBROUTINE (B0004 in BUFRKERN.FOR) - (FORT77)

PURPOSE Decodes a BUFR (Section 4) bit string

ARGUMENTS

DESCR	(Entry/exit)	I4	Array of descriptors (expanded list on exit)
VALUES	(Exit)	I4	NOBS*ND decoded values
NAMES	(Exit)	C*	Decoded characters
ND	(Entry/exit)	I4	Number of descriptors
NOBS	(Entry)	I4	Expected number of observations
MESAGE	(Entry)	C*	BUFR section 4 bit string
CMPRES	(Entry)	L4	TRUE (.T.) If BUFR compression
DUNIT	(Entry)	I4	Unit number for display (<=0 to suppress display)
IERR	(Exit)	I4	Error code (0=OK, 41x=error)

FUNCTION The general approach is that proposed by ECMWF: a single scan through a descriptor string whose still unscanned part will grow as sequence or replication descriptors are met. Each descriptor in the expanded string corresponds to 'NOBS' values in the array of numbers; a zero descriptor corresponds to "associated" (quality control) fields for the element which follows. Character fields are returned in the character string, with pointers rather than values in the array of numbers; descriptors in the expanded string are flagged if they correspond to characters and hence to pointers. Flags are also set for repeated co-ordinate increments to which no values correspond in the array. If the display flag is set, code figures are looked up in TABLE B; but the code figure itself is displayed if a description cannot be found. Plain language is just skipped unless a display is requested. Error return if BUFR rules broken in one of the ways below:
411 - no entry for descriptor in TABLE B or TABLE D
412 - operator descriptor (F=2), but no operation defined (X>5)
413 - delayed replication (Y=0), but no count in data
414 - delayed replication with compression, but counts vary
415 - new reference value(s), but end of definition not marked
416 - new reference values with compression, but values vary
417 - all values missing with compression, but increment width nonzero
418 - data repetition with more than one descriptor
419 - data repetition but compression error return points ND to descriptor in expanded string

SUBPROGRAMS ASC2EB, BVALUE, CODFLG, DESFXY, TABLEB, TABLED

INTRINSICS BTEST, LEN

CALLED FROM DEBUFR

REFERENCES 1) Manual on codes, WMO-306, Vol. 1-2, Part B, 1999 (with supplement No.3, August 1998)

15.

IDENTIFIER DECODE_FDP

TYPE MAIN (B2000 in DECODFDP.FOR) - (FORT77)

PURPOSE Control program to decode ERS fast delivery products

INCLUDES SYSTEM.FOR System definitions

FUNCTION Main level routine to decode ERS LBR products and output to separate product files.File name containing GTS bulletins and optional FDP file names are specified on the command line. On VAX/VMS systems the command must be set up as a 'foreign' command: e.g.

$ DECODE :== $disk:[dir]DECODFDP.EXE

$ DECODE [gtsfile] [uatfile] [urafile] [uwafile] [uwifile] [dmpfile]

where:

gtsfile is the input GTS/BUFR encoded data file,
uatfile is the output FDP file for UAT products,
urafile is the output FDP file for URA products,
uwafile is the output FDP file for UWA products,
uwifile is the output FDP file for UWI products,
dmpfile is the output dump file for the decoded data (recommended for diagnostic use only)

All file names are optional. If blank the GTS input file defaults to **FDP_BUFR.GTS** and FDP output files are created with names of the form **Uxx__ddhhmm.Ent** where **Uxx** is the product type, **ddhhmm** is the day, hour and minute of the first of that type of product in the GTS file (taken from the WMO header), **n** is 1 or 2 for ERS-1 or ERS-2 and **t** is blank for operationa data or T for test data. Note that if no data for any particular product is present in the GTS stream, then the corresponding FDP file will not be created. Error and Information messages are output to the standard output stream. The default, if **dmpfile** is blank, is no dump output.

SUBPROGRAMS CHCASE, DEBUFR_FDP, GETARG

FILES UNIT(*) FILE(SYS$OUTPUT) Messages (Output)

16.

IDENTIFIER	DESFXY			
TYPE	SUBROUTINE (B0013 in BUFRKERN.FOR) - (FORT77)			
PURPOSE	Decode a BUFR descriptor			
ARGUMENTS	DESCR	(Entry)	I4	BUFR descriptor
	F	(Exit)	I4	Descriptor type
	X	(Exit)	I4	Class/category/number of elements
	Y	(Exit)	I4	Class element/entry/operation
FUNCTION	Splits up a 16-bit BUFR descriptor (FXXYYY) into F (2 bits), X (6 bits) and Y (8 bits).			
INTRINSICS	MOD			
CALLED FROM	BUFRER, DEBUFR, DECOD4, ENBUFR			

17.

IDENTIFIER	DT2I			
TYPE	SUBROUTINE (G0201 in DATETIME.FOR) - (FORT77)			
PURPOSE	Convert date/time between text and integer formats			
ARGUMENTS	DTC	(Entry/exit)	C24	Date/time in text format as "DD-MMM-YYYY HH:MM:SS.TTT"
	DTI	(Entry/exit)	I4	7-element date (YYYY,MM,DD) and time (HH,MM,SS,TTT)
	IND	(Entry)	I4	Conversion direction indicator >=1 character to integer <=0 integer to character
FUNCTION	Converts date and time information between character form (for example "26-MAR-1990 11:51:44.123") and integer representation of each element (for example 1990,3,26,11,51,44,123). The direction of conversion depends on the value of 'IND'. Only the month is checked for correct value; an unknown 3-character month is returned as DTI(2) = 13, or '***' if DTI(3) is not 1-12. 21st century is assumed for year values 0 to 49, 20th century for year values 50 to 99 and Year 2000 for year value 100.			
CALLED FROM	ENBUFR_UWA, ENBUFR_UWI, ENBUFR_URA, ENBUFR_UAT, ERROUT, HDR_CF, LOGOUT, ORBPAR, UAT_CF, UAT_IO			

18.

IDENTIFIER	DT_NOW			
TYPE	SUBROUTINE (G0208 in DATETIME.FOR) - (VAXFORT)			
PURPOSE	Returns current system date/time as integers			
ARGUMENTS	DTI	(Exit)	I4	7-element current date (yyyy,mm,dd) and time (hh,mm,ss,ttt)

FUNCTION Gets current VAX system date and time as YR,MN,DY,HR,MN,SC,TH
N.B. For UNIX :-
If possible, replace the VAX 'IDATE' and 'SECNDS' calls with local equivalents, and if necessary modify the code so that the same information is passed back to the caller. If no date/time system interface is available, just set all the 'DTI' elements to zero. (For HP systems, compiling FORTRAN routines with option +E1 invokes a non-Year 2000 compliant IDATE routine; compiling with option +U77 gives a Year 2000 compliant IDATE routine but 'seconds' are not returned).

INTRINSICS IDATE, INT, NINT, SECNDS

CALLED FROM ENBUFR_UAT, ENBUFR_URA, ENBUFR_UWA, ENBUFR_UWI, ERROUT, LOGOUT

19.

IDENTIFIER	EB2ASC			
TYPE	SUBROUTINE (B0017 in BUFRKERN.FOR) - (FORT77)			
PURPOSE	Convert EBCDIC characters to ASCII			
ARGUMENTS	NC	(Entry)	I4	Number of characters to convert
	STRING	(Entry/Exit)	C*	Character string
FUNCTION	Converts first 'NC' characters of 'STRING' from EBCDIC to ASCII representation.			
CALLED FROM	DEBUFR, DEBUFR_FDP, ENBUFR, GTSHDR, NCHARS			

20.

IDENTIFIER	ERRCODES
TYPE	INCLUDE (I0002 in ERRCODES.FOR) - (FORT77)
PURPOSE	Include file for error code parameter definitions
INCLUDED IN	B2T_UAT, B2T_URA, B2T_UWA, B2T_UWI, DEBUFR_FDP, DEBUFR_UAT, DEBUFR_URA, DEBUFR_UWA, DEBUFR_UWI, ENBUFR_FDP, ENBUFR_UAT, ENBUFR_URA, ENBUFR_UWA, ENBUFR_UWI, ENCODE_FDP, ERROUT, GTS_IO, UAT_CF, URA_CF, UWA_CF, UWI_CF,

21.

IDENTIFIER ERSORB

TYPE SUBROUTINE (G0401 in ERSORB.FOR) - (FORT77)

PURPOSE Dummy ERSORB interface for end-users

ARGUMENTS

MODE	(Entry)	I4	Mode of operation -1,0 = initialise 1,2 = propagate
RTIME	(Entry/Exit)	R8	Time of ascending node (M.J.D.)
XM	(Entry/Exit)	R8	6-element mean Kepler state
PTIME	(Entry/Exit)	R8	Epoch (M.J.D.)
X	(Exit)	R8	6-element interpolated Kepler state
RR	(Entry/Exit)	R8	3-element state vector x,y,z position (km)
RRD	(Entry/Exit)	R8	3-element state vector x,y,z velocity (km/s)
RES	(Exit)	R8	42-element results vector
IERR	(Exit)	I4	4-element error return vector

INCLUDES CERSORB_BLOCK.FOR Dummy BLOCK DATA

FUNCTION This routine provides a dummy interface to the ESA 'ERSORB' package, which is *not* provided to end-users as part of the ERS BUFR distribution. The ERSORB routine is not called by the BUFR decoder, but may be required to satisfy the linker as it is indirectly referenced in common encode/decode modules. This dummy version is needed for use with the encoder for testing purposes only. The real ERSORB package is provided only to the BUFR software developer as 'contractor furnished equipment', and this dummy routine should *not* be used for development purposes. This dummy routine merely indicates an error condition in 'IERR' for all calls, of whatever mode, and sets the appropriate output parameters to zero, depending on whether called for initialise or propagate mode. Algorithm returns an error condition for all calls. Set RTIME and XM to zero for mode=-1,0 only; set PTIME, RR and RRD to zero for mode=1,2 only ; set RES and X to zero for all calls.

CALLED FROM ORBPAR

REFERENCES B Duesmann and H Klinkrad: "ERS Orbit Propagator Software (ERSORB). Interface and Installation Guide". ESA Document ER-TN-ESA-GS-0003 ISSUE 3.2, May 1994.

22.

IDENTIFIER	FDPTYPES
TYPE	INCLUDE (I0003 in FDPTYPES.FOR) - (VAXFORT)
PURPOSE	Include file for FDP product type parameter definitions
REFERENCES	ERS-2 kiruna station user interface specification. ER-IS-MDA-GS-2010, ISSUE 1/0, 5 APR 94
INCLUDED IN	B2T_UAT, B2T_URA, B2T_UWA, B2T_UWI, DEBUFR_FDP, DEBUFR_UAT, DEBUFR_URA, DEBUFR_UWA, DEBUFR_UWI, ENBUFR_FDP, ENBUFR_UAT, ENBUFR_URA, ENBUFR_UWA, ENBUFR_UWI, T2B_UAT, T2B_URA, T2B_UWA, T2B_UWI, UAT_IO, URA_IO, UWA_IO, UWI_IO

23.

IDENTIFIER	FRACTION_OF_DAY			
TYPE	FUNCTION (G0207 in DATETIME.FOR) - (FORT77)			
PURPOSE	Convert clock time to fraction of day			
ARGUMENTS	HR	(Entry)	I4	Hour (0-23)
	MI	(Entry)	I4	Minute (0-59)
	SE	(Entry)	I4	Second (0-59)
	TH	(Entry)	I4	1/1000s (0-999)
	FRACTION_OF_DAY	(Function)	R8	Fraction of day (0 - <1)
FUNCTION	Converts clock time (hour,minute,second and milli-seconds) to fraction of a day. N.B. Function must be declared double precision (REAL*8)			
CALLED FROM	JULIAN_TO_DATIM			

24.

IDENTIFIER	FXYDES			
TYPE	SUBROUTINE (B0014 in BUFRKERN.FOR) - (FORT77)			
PURPOSE	Encode a BUFR descriptor			
ARGUMENTS	F	(Entry)	I4	Descriptor type
	X	(Entry)	I4	Class/category/number of elements
	Y	(Entry)	I4	Class element/entry/operation
	DESCR	(Exit)	I4	BUFR descriptor
FUNCTION	Combines BUFR descriptor parts (F,X,Y format) into a coded BUFR descriptor.			
INTRINSICS	MOD			
CALLED FROM	ENBUFR, FDPSEQ			

25.

IDENTIFIER GETARG

TYPE SUBROUTINE (G0101 in UTILS.FOR) - (VAXFORT)

PURPOSE Get an argument from command line (VMS only)

ARGUMENTS

NARG	(Entry)	I4	Argument sequence number
CARG	(Exit)	C*	Argument string

FUNCTION Uses VAX/VMS system service routine 'LIB$GET_FOREIGN' to get a DCL command line, then extracts the NARGth argument. All arguments should be separated by one or more spaces or tabs on the DCL command line. Null arguments can be indicated by "" .All arguments are returned by LIB$GET_FOREIGN as upper case only unless in double quotes, which are removed by this routine. If NARG is greater than the number of given arguments the string is returned blank. The calling main program should be set up as a VMS 'foreign' command: e.g.

```
$ PROG :== $disk:[dir]PROG.EXE
$ PROG arg1 arg2 arg3 [...]
```

For example

```
$ PROG FILE1.DAT "" FILE3.DAT
```

N.B. For Unix platforms:-
1. This routine is intended to be reasonably compatible with intrinsic routine 'GETARG' common on many Unix Fortran implementations.
2. Unix users should remove or otherwise disable this routine from being linked in order that the intrinsic will be used.
3. Some Unix GETARGs may return the program name with NARG=1. In this case the calling routine will need to be modified to return the command line argumemts starting at NARG=2.

INTRINSICS INDEX

CALLED FROM DECODE_FDP, ENCODE_FDP

RTL LIB$GET_FOREIGN

26.

IDENTIFIER GETDIR

TYPE SUBROUTINE (G0103 in UTILS.FOR) - (FORT77)

PURPOSE Return a directory specification

ARGUMENTS

LOGDIR	(Entry)	C*	Directory name to translate: VAX/VMS: Logical name Unix: Environment variable
TRNDIR	(Exit)	C*	Translated directory spec.
NCD	(Exit)	I4	No. of significant chrs in TRNDIR

FUNCTION Uses GETENV, STRLEN
For VAX/VMS, returns the translation of the logical name contained in 'LOGDIR'; for Unix, returns the translation of the Environment Variable 'LOGDIR'. In both cases, the translation should be a valid directory specification for the local O/S, though a final '/' is appended to the Unix spec. if necessary. If the translation is null (blank), then the default '[]' (VAX/VMS) or './' (Unix) is returned to indicate the current default/working directory. Needs the Unix standard library routine 'GETENV' or the equivalent VAX emulator.

INTRINSICS INDEX, LEN

CALLED FROM FDPSEQ, IPBTAB, NEWTABLS

27.

IDENTIFIER GETENV

TYPE SUBROUTINE (G0104 in UTILS.FOR) - (VAXFORT)

PURPOSE Translate a logical name (VMS only)

ARGUMENTS

LOGNAM	(Entry)	C*	Logical name to be translated
TRNLOG	(Exit)	C*	Translated name

N.B. For Unix platforms:-
1. This routine is intended to mimic the Unix intrinsic routine 'GETENV' - common on many Unix Fortran implementations – which translates an environment variable.
2. Unix users should remove or otherwise disable this routine from being linked in order that the intrinsic will be used.

CALLED FROM GETDIR

RTL LIB$SYS_TRNLOG

28.

IDENTIFIER	GTS_IO			
TYPE	SUBROUTINE (B0501 in GTSIO.FOR) - (VAXFORT)			
PURPOSE	Input or output a complete GTS/BUFR message			
ARGUMENTS	GTSDSN	(Entry/exit)	C*	GTS file name or if 'CloseFile' closes currently open GTS file (default: 'FDP_BUFR.GTS')
	INPUT	(Entry)	L4	.T. for input else output
	GTSMSG	(Entry)	C*	GTS message string
	NBYTES	(Entry/exit)	I4	Number of bytes to write (on output) or read (on input)
	IERR	(Exit)	I4	Error code -1 = End of file (input) 0 = OK 601 = GTS message string too short (input) 898 = Input channel open error 899 = Input channel read error 998 = output channel open error 999 = output channel write error

COMMON	/BUFERR/			
	IEL	(Read)	I4	Index of element in error
	IOB	(Read)	I4	Index of observation in error
	IDES	(Read)	I4	Descriptor value in error
	IVAL	(Write)	I4	Observation value in error (indicates encode or decode for x01 errors or Main Product Header product type read for 702 errors or FORTRAN error code for Input/Output errors)

INCLUDES	ERRCODES.FOR	Error code definitions

FUNCTION	Read or write a complete GTS message from/to a given file. If the name is blank on the first call for either input or output the default name 'FDP_BUFR.GTS' is used and passed back to the calling routine. If this parameter contains the exact string 'CloseFile', the currently open file (for input or output depending on 'INPUT') is closed with no further action. For input the message string 'GTSMSG' should be declared to be the largest possible message length eg. 15000 bytes. File 'GTSDSN' is opened on first call and can be formatted or unformatted with variable or fixed length records. For output this version writes variable-length formatted records (depending on the length of the passed string 'message') to given disk file, which is opened on first entry with the specified name. Each record (message) is prefixed with four null bytes, required by the ESRIN X.25 transmitter software.
SUBPROGRAMS	STRLEN
INTRINSICS	CHAR, LEN
CALLED FROM	DEBUFR_FDP, ENBUFR_FDP

FILES	UNIT(11)	FILE(GTSDSN)	GTS message file	(Input)
	UNIT(12)	FILE(GTSDSN)	GTS message file	(Output)

29.

IDENTIFIER	HDR_CF			
TYPE	SUBROUTINE (B0705 in CONVFMT.FOR) - (VAXFORT)			
PURPOSE	Convert header parameters between RAL/ESA and BUFR			
ARGUMENTS	SATID	(Entry/Exit)	I4	Satellite identifier (1 or 2 for ERS-1/2)
	SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
	ORGID	(Entry/Exit)	I4	Originating centre Identifier (253 for ESRIN)
	STNID	(Entry/Exit)	I4	Station identifier (e.g. 1 for Kiruna)
	PCTRK	(Entry/Exit)	I4	Satellite track from north: 0-360 (degrees*1000)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector: (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (M*100) (4)-(6) = x,y,z velocity (M/S*100000)
	INSID	(Entry)	I4	Instrument identifier (e.g. 64 for RA)
	DTC	(Entry/Exit)	C24	(NO) Date and time strings: (DD-MMM-YYYY HH:MM:SS.TTT)
	CLAT	(Entry/Exit)	I4	(NO) Cell latitudes: -90 TO +90 (DEG*1000)
	CLON	(Entry/Exit)	I4	(NO) Cell longitudes: 0-360 (DEG*1000)
	NE	(Entry)	I4	First dimension of CELDAT (total number of elements)
	NO	(Entry)	I4	Second dimension of CELDAT (number of observations/cells
	CELDAT	(Exit/Entry)	I4	Array of (NE*NO) BUFR-format values common to all products:

Element	Parameter
1	Satellite Identifier (1/2 for ERS-1/2)
2	Software Identifier (e.g. 321)
3	Origin Identifier (253 for ESRIN)
4	Station Identifier (e.g. 1 for Kiruna)
5	Track direction (deg)

			6-11	UTC time of ascending node (YR, MTH, DAY, HR, MIN, SEC*1000)
			12-14	Satellite. x,y,z position (m*100)
			15-17	Satellite x,y,z velocity (m/s*10**5)
			18	Instrument Identifier (e.g. 64 for Radar Altimeter)
			19-24	UTC time of observation (YR, MTH, DAY, HR, MIN, SEC*1000)
			25	Observation latitude (deg*100)
			26	Observation longitude (deg*100)
	LE	(Exit)	I4	Last element used (26)
	IND	(Entry)	I4	Direction of conversion: >= 1 : To BUFR Format <= 0 : To ESA Format

INCLUDES

ERRCODES.FOR	Error code definitions
FDPTYPES.FOR	FDP product and BUFR data types
STATIONS.FOR	Ground stations definitions

FUNCTION Converts common header parameters between RAL/ESA format (see Refs 1 and 2) and BUFR format (Ref 3). Direction of conversion controlled by value of 'IND'. The BUFR format parameters are contained in the integer array 'CELDAT'. Elements 1-24 are identical for all observations; elements 25 and 26 are converted as given. All other elements are unused/undefined.

SUBPROGRAMS DT2I, ORBPAR, STATIONS

INTRINSICS FLOAT, NINT

CALLED FROM UAT_CF, URA_CF, UWA_CF, UWI_CF

REFERENCES
1) P.Bailey. SADIST-2 V100 Products. RAL, ER-TN-RAL-AT-2164, 6 September 1995..
2) ERS-2 KIRUNA Station User Interface Specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 94
5) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

30.

IDENTIFIER IARGC

TYPE FUNCTION (G0101 in UTILS.FOR) - (VAXFORT)

PURPOSE Return a count of arguments on command line (VMS only)

ARGUMENTS IARGC (Function) I4 No. of arguments

FUNCTION Uses (VMS) utility routine GETARG. The calling main program should be set up as a VMS 'foreign' command: e.g.

```
$ PROG :== $disk:[dir]PROG.EXE
$ PROG arg1 arg2 arg3 [...]
```

For example

```
$ PROG FILE1.DAT "" FILE3.DAT
```

N.B. For Unix platforms:-
1. This routine is intended to be reasonably compatible with intrinsic routine 'IARGC' common on many Unix Fortran implementations.
2. Unix users should remove or otherwise disable this routine from being linked in order that the intrinsic will be used. Howver, if IARGC is not available, but GETARG is, this routine can be used.

CALLED FROM DECODE_BUFR

31.

IDENTIFIER IPBTAB

TYPE SUBROUTINE (B0008 in BUFRKERN.FOR) - (FORT77)

PURPOSE Input BUFR tables

ARGUMENTS NONE

COMMON / BUFTAB / LOADED
/ BUFRAC / CATEG
/ BUFRBC / KERNELB, ELMENT, LOCALB
/ BUFRBI / CLASS
/ BUFRDC / KERNELD, SEQPTR, LOCALD
/ BUFRDI / CATPTR
/ BUFRCC / CFTEXT
/ BUFRCI / NCODES, INDEX

FUNCTION Reads in BUFR tables B and D and code table, initialising appropriate common blocks. This routine should be called once only, before any encoding or decoding routines. The files are opened and closed internally to this routine.
Input files are looked for in the directory pointed to by 'BUFRDIR' - a pre-defined logical name under OpenVMS or an envirinment variable under Unix. The default is the current working directory in both cases. Table B and Table D are required for both encoder and decoder operation, Codeflg for decoder; if any of these cannot be found or opened, the program will be stopped. Table A is not critical and the program will continue.

FILES	UNIT(51)	FILE(BUFTABLA.DAT)	Table A (input)
	UNIT(52)	FILE(BUFTABLB.DAT)	Table B (input)
	UNIT(53)	FILE(BUFTABLD.DAT)	Table D (input)
	UNIT(54)	FILE(BUFCODFG.DAT)	Code table (input)

CALLED FROM CODFLG, TABLEB, TABLED

32.

IDENTIFIER	JULIAN_DATE			
TYPE	SUBROUTINE (G0204 DATETIME.FOR) - (FORT77)			
PURPOSE	CONVERT JULIAN DATE TO CALENDAR DATE			
ARGUMENTS	JDT	(Entry)	R8	Julian date/time
	YR	(Exit)	I4	Year (>0)
	MO	(Exit)	I4	Month (1-12)
	DY	(Exit)	I4	Day (1-31)
FUNCTION	Converts Julian date/time (in format dddddd.ffffff) to calendar date - i.e. year, month and day. Allows for half-day offset. N.B. JDT argument must be declared double precision (REAL*8).			
INTRINSICS	INT			
CALLED FROM	JULIAN_TO_DATIM			
REFERENCES	1) Astronomical Formulae for Calculators, Jean Meeus, Pages 23-25.			

33.

IDENTIFIER JULIAN_OF_YEAR

TYPE FUNCTION (G0203 in DATETIME.FOR) - (FORT77)

PURPOSE Calculate Julian date for 1 January of given year

ARGUMENTS

YR	(Entry)	I4	Full year (>=0)
JULIAN_OF_YEAR	(Function)	R8	Julian date/time of 1-January of the year

FUNCTION Calculates the Julian date at midnight, 1st January of given year. Since Julian days begin at midday, the returned value is 0.5 less than the calendar equivalent. Year must include century and be A.D.
N.B. Function must be declared double precision (REAL*8).

JULIAN PERIOD:
Number of consecutive days since 1-January-4713 B.C.
Devised by Joseph Fustus Scaliger in 1582, named after his father, Julius Caesar Scaliger. Based on 'cycles of history' of period 7980 years, being the product of:
28 = so-called solar cycle in Julian calendar
19 = lunar or metonic cycle
15 = 'cycle of indiction', an ancient roman taxation period.
4713 bc was the nearest past year on which all three cycles started together. So there!

Algorithm:- Calculate number of days since 1-January-0000 A.D, to 1-January-year, allowing for leap years and centuries, and add offset for base date (1-January-4713 B.C), less half a day.

CALLED FROM JULIAN_TO_DATIM

REFERENCES
1) Astronomical Formulae for Calculators, Jean Meeus, pages 23-25.
2) Encyclopaedia Brittanica, Micropaedia Vol.V, 15th ed. 1975.

34.

IDENTIFIER JULIAN_TIME

TYPE SUBROUTINE (G0205 in DATETIME.FOR) - (FORT77)

PURPOSE Convert Julian time to clock time

ARGUMENTS

JDT	(Entry)	R8	Julian date/time
HR	(Exit)	I4	Hour (0-23)
MN	(Exit)	I4	Minute (0-59)
SE	(Exit)	I4	Second (0-59)
TH	(Exit)	I4	1/1000S (0-999)

FUNCTION Converts Julian date/time (in format dddddd.ffffff) to clock time - i.e. hour, minute, second and milli-seconds. Allows for half-day offset. N.B. JDT argument must be declared double precision (REAL*8).

INTRINSICS INT, MOD

CALLED FROM JULIAN_TO_DATIM

35.

IDENTIFIER JULIAN_TO_DATIM

TYPE SUBROUTINE (G0202 in DATETIME.FOR) - (FORT77)

PURPOSE Convert between Julian and calendar date/time

ARGUMENTS

JDT	(Entry/Exit)	R8	Julian date/time
IDT	(Entry/Exit)	7I4	Calender date & clock time (Year,Month,Day,Hour,Minute, Second,Milli-seconds)
INV	(Entry)	I4	Conversion flag: >0 : JDT to IDT <=0 : IDT to JDT

FUNCTION Converts between date/time of Julian (in format dddddd.ffffff) to calendar date and time (year,month,day,hour,minute,second and milli-seconds). INV indicates direction of conversion: >0 to convert Julian to calendar else vice-versa.
N.B. JDT argument must be declared double precision (REAL*8).

Algorithm:- INV >0 : Convert Julian day number to year, month and day, convert fraction of day to hour, minute, second and milli-seconds.
INV <1 : Convert year, month & day to days-in-year. Calculate Julian day for 1 January in that year, add daynumber and fraction of day from hour, minute, second and milli-seconds. 21st century is assumed for year value 0 to 49, 20th century for year value 50 to 99 and Year 2000 for year value 100.

SUBPROGRAMS JULIAN_OF_YEAR, JULIAN_DATE, JULIAN_TIME, DAY_OF_YEAR, FRACTION_OF_DAY

CALLED FROM ENBUFR_UAT, ENBUFR_URA, ENBUFR_UWA, ENBUFR_UWI, ORBPAR, UAT_IO

36.

IDENTIFIER	OPEN_FDP			
TYPE	FUNCTION OPEN_FDP (G0107 in UTILS.FOR) - (FORT77)			
PURPOSE	Open an FDP file for read or write			
ARGUMENTS	IUNIT	(Entry)	I4	Unit Number
	CNAME	(Entry)	C*	File name
	LINPUT	(Entry)	L4	Input/output flag .T. = input file (file must exist) .F. = output file (file is created)
	IERR	(Exit)	I4	Error code
FUNCTION	Opens an FDP file for input (read existing file only) or output (creates a new file). Input/output status is returned.			
FILES	Unit (IUNIT)	File (CNAME)		FDP file (input or output)
CALLED FROM	DEBUFR_FDP			
REFERENCES	ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 94			

37.

IDENTIFIER ORBPAR

TYPE SUBROUTINE (G0301 in ORBITS.FOR) - (FORT77)

PURPOSE Calculate orbit parameters from state vector

ARGUMENTS

DTC	(Entry/Exit)	C24	Date and time for required orbit parameters (DD-MMM-YYYY HH:MM:SS.TTT)
DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
OPAR	(Exit)	R4	5-element orbit parameters (1) = SSP latitude (deg) (2) = SSP longitude (deg) (3) = satellite altitude above datum (km) (4) = satellite ground speed (km/s) (5) = satellite ground track (degrees true)
IERR	(Exit)	I4	Exit status 0 = OK 1 = error in initialising 2 = error in propagation

FUNCTION Uses ESA-supplied routines (via ERSORB interface) to propagate orbit - given a state vector and its epoch - from last ascending node to that at required time, and thence to calculate and extract various parameters such as satellite latitude/longitude/altitude and ground speed/track for return. On exit, IERR indicates the status of the calculation; if an error occurs, all elements of OPAR will be set to zero. For a list of possible internal errors, see reference.
Algorithm:-
1) convert ascending node date/time string to modified julian day
2) initialise propagator with state vector at this epoch
3) convert required epoch date/time string to modified julian day
4) propagate orbit to this epoch
5) extract required orbit parameters for return

SUBPROGRAMS DT2I, ERSORB, JULIAN_TO_DATIM

CALLED FROM HDR_CF, UWA_CF

REFERENCES B Duesmann and H Klinkrad: "ERS Orbit Propagator Software

(ERSORB). Interface and Installation Guide". ESA Document ER-TN-ESA-GS-0003 Issue 3.2, May 1994.

38.

IDENTIFIER STATIONS

TYPE INCLUDE (I0004 in STATIONS.FOR) - (FORT77)

PURPOSE INCLUDE file for ground station definitions

FUNCTION Look-up-tables to convert between FDP "ground station" and BUFR table definitions. Conversion as follows:

Station	FDP	BUFR
Kiruna	1	1
Fuchino	2	2
Gatineau	3	3
Maspolomas	4	4
EECF (Frascati)	5	5
Prince Albert	6	6
Tromso	13	13

The above definitions are those in code table 001034. In addition, the ICAO and BUFR 'master' table (001033) for ESA/ESRIN are defined here.

CALLED FROM HDR_CF

REFERENCES ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 94

39.

IDENTIFIER STRLEN

TYPE FUNCTION STRLEN (B0018 in BUFRKERN.FOR) - (FORT77)

PURPOSE Find position of last character in a string.

ARGUMENTS				
	STRING	(ENTRY)	C*	Character string.
	STRLEN	(FUNCN)	I4	Index position of last character.

FUNCTION 'STRLEN' returns the number of significant characters in 'STRING'. 'STRING' is assumed to be left justified. The total length of 'STRING' is found and 'STRLEN' is the index position of the first non-'space' character from the right. Control characters are treated as spaces. STRLEN returns zero if there are no non-space characters in the string.

Algorithm: starting at right-most character, convert it to ASCII (on EBCDIC machines); if ASCII equivalent value is <=32 (' ') or >= 127 ('~'), then return current position in string, else loop to next character to left. Stop at left-most character, setting count to zero if this last character also fails the test.

SUBPROGRAMS EB2ASC

INTRINSICS ICHAR, LEN

CALLED FROM DEBUFR, ENBUFR_FDP, ERROUT, GTS_IO

40.

IDENTIFIER TABLEA

TYPE SUBROUTINE (B0009 in BUFRKERN.FOR) - (FORT77)

PURPOSE Look up the data category in TABLE A

ARGUMENTS				
	VALUE	(Entry)	I4	Code value (0-255)
	WORDS	(Exit)	C*	Data category string

COMMON / BUFTAB / LOADED
/ BUFRAC / CATEG

FUNCTION Given a code value 0-255, returns the corresponding data category description from TABLE A. Codes not yet defined are returned as 'reserved'; values out of range are returned 'invalid code value'. The BUFR tables are loaded automatically if not already in memory.

SUBPROGRAMS IPBTAB

INTRINSICS ICHAR

CALLED FROM DEBUFR

41.

IDENTIFIER TABLEB

TYPE SUBROUTINE (B0010 in BUFRKERN.FOR) - (FORT77)

PURPOSE Look up the details of an element in TABLE B

ARGUMENTS

X	(Entry)	I4	'X' part of descriptor
Y	(Entry)	I4	'Y' part of descriptor
SCALE	(Exit)	I4	Scale value (power of 10)
REFVAL	(Exit)	I4	Reference value
WIDTH	(Exit)	I4	Data width in bits
FORMAT	(Exit)	C1	Format type ('F','C','R')
NAME	(Exit)	C60	Name of element
UNITS	(Exit)	C24	Units of element
IERR	(Exit)	I4	0=OK, 1=not in table

COMMON / BUFTAB / LOADED
/ BUFRBC / KERNELB, ELMENT, LOCALB
/ BUFRBI / CLASS

FUNCTION Given the 'X' and 'Y' parts of a BUFR element descriptor, looks up the element scaling, reference value, data width, format type, name and units from BUFR TABLE B. If the element is not defined in TABLE B, 'IERR' is returned non-zero. The BUFR tables are loaded automatically if not already in memory.

SUBPROGRAMS IPBTAB

INTRINSICS ICHAR

CALLED FROM DECOD4, ENBUFR, ENCOD4

42.

IDENTIFIER TABLED

TYPE SUBROUTINE (B0011 in BUFRKERN.FOR) - (FORT77)

PURPOSE Look up a sequence of descriptors in TABLE D

ARGUMENTS

X	(Entry)	I4	'X' part of sequence descriptor
Y	(Entry)	I4	'Y' part of sequence descriptor
SEQ	(Exit)	I4	Array of descriptors in sequence
NSEQ	(Exit)	I4	Number of entries in sequence (0-16)

COMMON / BUFTAB / LOADED
/ BUFRDC / KERNELD, SEQPTR, LOCALD
/ BUFRDI / CATPTR

FUNCTION In the "kernel", the most frequently used eighth of the table, a TABLE D entry consists of a count followed by the descriptors in the sequence; the Y of the descriptor itself comes between count and expansion in the "local" section of the table. 'NSEQ' is returned as zero if descriptor not found. The BUFR tables are automatically loaded if not already in memory.

SUBPROGRAMS IPBTAB

INTRINSICS ICHAR

CALLED FROM DECOD4, ENBUFR

43.

IDENTIFIER	UAT_CF			
TYPE	SUBROUTINE (B0701 in CONVFMT.FOR) - (VAXFORT)			
PURPOSE	Convert UAT parameters between RAL/ESA and BUFR formats			
ARGUMENTS	SATID	(Entry/Exit)	I4	Satellite identifier (1 or 2 for ERS-1/2)
	SWID	(Entry/Exit)	I4	Software identifier (e.g. 200 for SADIST-2)
	ORGID	(Entry/Exit)	I4	Originating centre Identifier (253 for ESRIN)
	STNID	(Entry/Exit)	I4	Station identifier (e.g. 13 for Tromso)
	PCTRK	(Entry/Exit)	I4	Satellite track from north, 0-360 (deg*1000)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
	DTC	(Entry/Exit)	C24	150 date and time strings (DD-MMM-YYYY HH:MM:SS.TTT)
	CLAT	(Entry/Exit)	I4	150 cell latitudes, -90 TO +90 (deg*1000)
	CLON	(Entry/Exit)	I4	150 cell longitudes, 0-360 (deg*1000)
	ATBN	(Entry/Exit)	I2	150 across-track band numbers (0-9)
	NSST	(Entry/Exit)	I2	10*150 sea surface temperatures (deg.K*100) (nadir only view)
	DSST	(Entry/Exit)	I2	10*150 sea surface temperatures (deg.K*100) (dual-view)
	PCD	(Entry/Exit)	I4	150 product confidence data (cell)
	CELDAT	(Exit/Entry)	I4	Array of 64*150 BUFR values
	IND	(Entry)	I4	Direction of conversion: >= 1 : to BUFR format <= 0 : to ESA format

INCLUDES	ERRCODES.FOR	Error code definitions
	FDPTYPES.FOR	FDP product and BUFR data types

FUNCTION Converts UAT parameters between RAL/ESA format (see Refs 1 and 2) and BUFR format (Ref 3). Direction of conversion controlled by value of 'IND'. The BUFR format parameters are contained in the integer array 'CELDAT'.

SUBPROGRAMS HDR_CF

INTRINSICS BTEST, FLOAT, IBSET, MOD, NINT

CALLED FROM B2T_UAT, DEBUFR_UAT, ENBUFR_UAT, T2B_UAT

REFERENCES
1) P. BAILY: SADIST-2 V100 PRODUCTS. ER-TN-RAL-AT-2164, 6 September 1995.
2) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 April 1994
3) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, May 1999

44.

IDENTIFIER	UAT_IO			
TYPE	SUBROUTINE (U0001 in FDPIO.FOR) - (VAXFORT)			
PURPOSE	ERS-1/2 UAT product input/output routine (SADIST-2)			
ARGUMENTS	UNIT	(Entry)	I4	File unit number
	INPUT	(Entry)	L4	TRUE (.T.) for input, FALSE (.F.) for output
	SATID	(Entry/exit)	I4	Satellite identifier (1 for ERS-1, 2 for ERS-2)
	SWID	(Entry/exit)	I4	Software identifier (e.g. 200 for SADIST-2)
	STNID	(Entry/exit)	I4	Station identifier (e.g. 13 for Tromso)
	PRDID	(Exit)	I4	Product identifier (23 for UAT)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
	DTC	(Entry/exit)	C24	150 date and time strings (DD-MMM-YYYY HH:MM:SS.TTT)
	PCPCD	(Entry/exit)	I2	Product confidence data (Processing quality)
	PCLAT	(Entry/exit)	I4	Product centre latitude: -90 to +90 (deg*1000)
	PCLON	(Entry/exit)	I4	Product centre longitude: 0 to 360 (deg*1000)
	PCTRK	(Entry/exit)	I4	Satellite track from north: 0 to 360 (deg*1000)
	CLAT	(Entry/exit)	4	150 cell latitudes: -90 to +90 (deg*1000)
	CLON	(Entry/exit)	I4	150 cell longitudes: 0 to 360 (deg*1000)
	ATBN	(Entry/exit)	I2	150 across-track band numbers (0-4)
	NSST	(Entry/exit)	I2	10*150 sea surface temperatures (deg.k*100) (nadir only view) (1*Mean 30-arcmin+9*10-arcmin)
	DSST	(Entry/exit)	I2	10*150 sea surface temperatures (deg.k*100) (dual view) (1*Mean 30-arcmin+9*10-arcmin)

	PCD	(Entry/exit)	I4	150 product confidence data (cell)
	IERR	(Exit)	I4	Error/status code <0 = end of file (for input) 0 = OK >0 = read or write error
	BLANK	(Exit)	L4	.T. if product is blank/invalid
COMMON	/ BUFERR /			
	IEL	(Write)	I4	Number of non-dummy cells in this sub-product
	IOB	(Write)	I4	Accumulative number of cells read from UAT product
	IDES	(Write)	I4	Total number of cells in the UAT product
	IVAL	(Write)	I4	1 if last sub-product, otherwise 0

INCLUDES FDPTYPES.FOR Fast Data Products identifier definitions

FUNCTION Reads or writes a nominal (SADIST-2 format) UAT sub-product of 150 cells from/to file. For compatibility with ISS, the file must have been previously opened with the keywords FORM='UNFORMATTED', RECL=4096, RECORDTYPE='VARIABLE' and CARRIAGECONTROL='LIST' on unit 'UNIT'. The RECL and CARRIAGECONTROL keywords are optional for reading but must be used when opening a file for writing. *** *Note On VAX systems RECL is in 4-byte longwords, so the maximum record length is 16384 bytes****.

Although the ASST 'product' is data for one whole orbit (i.e one tape recorder dump), this routine returns a sub-product of the next 150 cells from the input file. RAL parameters for date/time, latitude/longitude are converted to ESA format (Ref 2). If there are not a multiple of 150 cells remaining, the sub-product is padded with dummy data (SST's etc. set to missing data values). There may be more than one product in the input file as long as each product starts a new record. On output the file may contain one or more sub=products, each being one record.

On exit after a read operation, COMMON /BUFERR/ parameters are set thus:-

IEL = Number of valid (non-dummy) cells returned for that sub-product. This is normally 150 but may be less for the last sub-product.

IOB = Accumulated number of cells read from the product.

IDES = Total number of cells in the product.

IVAL = 1 for the last sub-product, =0 otherwise.

If IVAL=1 and IOB<IDES, this indicates that the UAT file contains more cells than can be handled by this routine, and the product will be truncated. another call to this routine will begin reading a new product or indicate an end of file if there are no more products.

SUBPROGRAMS DT2I, JULIAN_TO_DATIM

INTRINSICS CHAR, ICHAR, INDEX, INT, MOD, NINT

CALLED FROM B2T_UAT, DEBUFR_UAT, ENBUFR_UAT, T2B_UAT

REFERENCES
1) P.Bailey. SADIST-2 V100 products. ER-TN-RAL-AT-2164, 6 September 1995.
2) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.

45.

IDENTIFIER URA_CF

TYPE SUBROUTINE (B0702 in CONVFMT.FOR) - (VAXFORT)

PURPOSE Convert URA parameters between ESA and BUFR formats

ARGUMENTS	SATID	(Entry/Exit)	I4	Satellite identifier (1 or 2 for ERS-1/2)
	SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
	ORGID	(Entry/Exit)	I4	Originating centre Identifier (253 for ESRIN)
	STNID	(Entry/Exit)	I4	Station identifier (e.g. 13 for Tromso)
	PCTRK	(Entry/Exit)	I4	Satellite track from north, 0-360 (deg*1000)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
	DTC	(Entry/Exit)	C24	77 date and time strings (DD-MMM-YYYY HH:MM:SS.TTT)
	CLAT	(Entry/Exit)	I4	77 cell latitudes, -90 to +90 (deg*1000)
	CLON	(Entry/Exit)	I4	77 cell longitudes, 0-360 (deg*1000)
	SPD	(Entry/Exit)	I2	77 wind speeds (m/s*100)
	SDS	(Entry/Exit)	I2	77 standard.deviation wind speed (m/s*10000)
	SWH	(Entry/Exit)	I2	77 significant wave height (m*100)
	SDH	(Entry/Exit)	I2	77 standard.deviation SWH (m*10000)
	ALT	(Entry/Exit)	I4	77 altitudes (m*100)
	SDA	(Entry/Exit)	I4	77 standard.deviation altitude (m*10000)
	NBA	(Entry/Exit)	I2	77 number of blocks in average
	AVP	(Entry/Exit)	I2	77 average peakiness (*100)
	AVS	(Entry/Exit)	I2	77 average backscatter (db*100)
	IED	(Entry/Exit)	I2	77 integrated electron density ($\log(m^{-2})$*1000)
	CAL	(Entry/Exit)	I1	77 calibration status
	IMD	(Entry/Exit)	I1	77 instrument mode

ACOR	(Entry/Exit)	I4	4*77 altitude corrections (m*1000) (1) = ionosphere (2) = wet troposphere (3) = dry troposphere (4) = calibration constant
CCOR	(Entry/Exit)	I4	2*77 calibration corrections (m*1000) (1) = open loop htl (2) = open loop agc
PCD	(Entry/Exit)	I1	77 product confidence data (cell)
CELDAT	(Exit/Entry)	I4	Array of 45*77 scaled values
IND	(Entry)	I4	Direction of conversion: >= 1 : to BUFR format <= 0 : to ESA format

INCLUDES

ERRCODES.FOR	Error code definitions
FDPTYPES.FOR	FDP product and BUFR data types

FUNCTION Converts URA parameters between ESA format (see Ref 1) and BUFR format (Ref 2). Direction of conversion controlled by value of 'IND'. The BUFR format parameters are contained in the integer array 'CELDAT'.

SUBPROGRAMS HDR_CF, DT2I

INTRINSICS FLOAT, MAX, MOD, NINT

CALLED FROM B2T_URA, DEBUFR_URA, ENBUFR_URA, T2B_URA

REFERENCES 1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 94
2) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

46.

IDENTIFIER URA_IO

TYPE SUBROUTINE (U0002 in FDPIO.FOR) - (VAXFORT)

PURPOSE ERS-1/2 URA product input/output routine

ARGUMENTS

UNIT	(Entry)	I4	File unit number.
INPUT	(Entry)	L4	TRUE (.T.) for input, FALSE (.F.) for output
SATID	(Entry/exit)	I4	Satellite identifier (1 for ERS-1, 2 for ERS-2)
SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
STNID	(Entry/exit)	I4	Station identifier (eg. 1 for Kiruna)
PRDID	(Exit)	I4	Product identifier (11 for URA)
DTV	(Entry/exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
SVEC	(Entry/exit)	I4	6-element ascending node state vector (1)-(3) = X,Y,Z position (m*100) (4)-(6) = X,Y,Z velocity (m/s*100000)
DTC	(Entry/exit)	C24	77 date and time strings (DD-MMM-YYYY HH:MM:SS.TTT)
PCPCD	(Entry/exit)	I2	Product confidence data (Processing quality)
PCLAT	(Entry/exit)	I4	Product centre latitude: -90 to +90 (deg*1000)
PCLON	(Entry/exit)	I4	Product centre longitude: 0 to 360 (deg*1000)
PCTRK	(Entry/exit)	I4	Satellite track from north: 0 to 360 (deg*1000)
CLAT	(Entry/exit)	I4	77 cell latitudes: -90 to +90 (deg*1000)
CLON	(Entry/exit)	I4	77 cell longitudes: 0 to 360 (deg*1000)
SPD	(Entry/exit)	I2	77 wind speeds (m/s*100)
SDS	(Entry/exit)	I2	77 standard deviation wind speed (m/s*10000)
SWH	(Entry/exit)	I2	77 significant wave height (m*100)
SDH	(Entry/exit)	I2	77 standard deviation significant wave height (m*10000)
ALT	(Entry/exit)	I4	77 altitudes (m*100)
SDA	(Entry/exit)	I4	77 standard deviation altitudes (m*10000)
NBA	(Entry/exit)	I2	77 number of blocks in average

PCD	(Entry/exit)	I1	77 product confidence data (cell)
AVP	(Entry/exit)	I2	77 average peakiness (*100)
AVS	(Entry/exit)	I2	77 average back scatterometer (db*100)
IED	(Entry/exit)	I2	77 integrated electron density (log(m**-2) *1000)
CAL	(Entry/exit)	I1	77 calibration status
IMD	(Entry/exit)	I1	77 instrument mode
ACOR	(Entry/exit)	I4	4*77 altitude corrections (m*1000) (1) = ionosphere (2) = wet troposphere (3) = dry troposphere (4) = calibration constant
CCOR	(Entry/exit)	I4	2*77 calibration corrections (m*1000) (1) = open loop height time lag (2) = open loop automatic gain control
IERR	(Exit)	I4	Error/status code <0 = end of file (for input) 0 = OK >0 = read or write error
BLANK	(Exit)	L4	.T. if product is blank/invalid

INCLUDES FDPTYPES.FOR FDP product and BUFR data types

FUNCTION Reads or writes a nominal URA product from/to file. For compatibility with ISS the file must have been previously opened with keywords FORM='UNFORMATTED', RECORDTYPE='VARIABLE', RECL=4096 and CARRIAGECONTROL='LIST' on unit 'UNIT'. The RECL and CARRIAGECONTROL keywords are optional for reading but must be used when opening a file for writing. *** *Note On VAX systems RECL is in 4-byte longwords, so the maximum record length is 16384 bytes****.
The file format is compatible with the ESA URA product specification, though not all the Main or Specific Product Header parameters are given values on output.

INTRINSICS BTEST, IBCLR, IBSET

CALLED FROM B2T_URA, DEBUFR_URA, ENBUFR_URA, T2B_URA

REFERENCES 1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.

47.

IDENTIFIER	UWAFIXED
TYPE	INCLUDE (I0001 in UWAFIXED.FOR) - (VAXFORT)
PURPOSE	Include file for UWA fixed parameters
FUNCTION	Provides nominal fixed values for UWA products, either not present in FDP file specifications, or as defaults (e.g. if the state vector is not valid).
INCLUDED IN	UWA_CF
REFERENCES	UWA processing algorithm specification 2.0. ER-TN-ESA-GS-0342, Issue 1.0, Nov 95.

48.

IDENTIFIER UWA_CF

TYPE SUBROUTINE (B0703 in CONVFMT.FOR) - (VAXFORT)

PURPOSE Convert UWA parameters between ESA and BUFR formats

ARGUMENTS	SATID	(Entry/Exit)	I4	Satellite identifier (1 or 2 for ERS-1/2)
	SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
	ORGID	(Entry/Exit)	I4	Originating centre Identifier (253 for ESRIN)
	STNID	(Entry/Exit)	I4	Station identifier (e.g. 13 for Tromso)
	PCTRK	(Entry/Exit)	I4	Satellite track from north, 0-360 (deg*1000)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
	DTC	(Entry/Exit)	C24	Date and time string (DD-MMM-YYYY HH:MM:SS.TTT)
	CLAT	(Entry/Exit)	I4	Product centre latitude, -90 to +90 (deg*1000)
	CLON	(Entry/Exit)	I4	Product centre longitude, 0-360 (deg*1000)
	INCID	(Entry/Exit)	I2	Incidence angle (deg*1000)
	CLUTTER	(Entry/Exit)	I4	Clutter noise estimate (*1000)
	RANRES	(Entry/Exit)	I4	Range resolution (pixel spacing), 0-1000 (m*1000)
	AZIRES	(Entry/Exit)	I4	Azimuth resolution (pixel spacing), 0-1000 (m*1000)
	MSCBN	(Entry/Exit)	I4	Maximum spectral component before normalisation
	SPIN	(Entry/Exit)	I1	12X12 spectral intensity values
	PCPCD	(Entry/Exit)	I2	Product confidence data (process)
	NISRA	(Entry/Exit)	I4	Number of image samples in range and azimuth ($NR+NA*2^{16}$)
	AZICCO	(Entry/Exit)	I4	Azimuth clutter cut-off

GT731M	(Entry/Exit)	5I4	Spectrum at wavelengths > 731m (1) Total image spectral energy (2) Mean wavelength (m) (3) Wavelength spread (m) (4) Mean direction (deg) (5) Direction spread (deg)
CELDAT	(Exit/Entry)	I4	Array of 349 BUFR values
IND	(Entry)	I4	Direction of conversion: >= 1 : to BUFR format <= 0 : to ESA format

INCLUDES

ERRCODES.FOR	Error code definitions
FDPTYPES.FOR	FDP product and BUFR data types
UWAFIXED.FOR	UWA 'fixed' parameters

FUNCTION Converts UWA parameters between ESA format (see Ref 1) and BUFR format (Ref 2). Direction of conversion controlled by value of 'IND'. The BUFR format parameters are contained in the integer array 'CELDAT'.

SUBPROGRAMS HDR_CF, ORBPAR

INTRINSICS BTEST, FLOAT, IBSET, MIN, MOD, NINT

CALLED FROM B2T_UWA, DEBUFR_UWA, ENBUFR_UWA, T2B_UWA

REFERENCES
1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 94
2) UWA processing algorithm specification 2.0. ER-TN-ESA-GS-0342, Issue 1.0, Nov 95.
3) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

49.

IDENTIFIER	UWA_IO			
TYPE	SUBROUTINE (U0003 in FDPIO.FOR) - (VAXFORT)			
PURPOSE	ERS-1/2 UWA product input/output routine			
ARGUMENTS	UNIT	(Entry)	I4	File unit number
	INPUT	(Entry)	L4	TRUE (.T.) for input, FALSE (.F.) for output
	SATID	(Entry/exit)	I4	Satellite identifier (1 for ERS-1, 2 for ERS-2)
	SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
	STNID	(Entry/exit)	I4	Station identifier (eg. 1 for Kiruna)
	PRDID	(Exit)	I4	Product identifier (5 for UWA)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
	DTC	(Entry/exit)	C24	Date and time string (DD-MMM-YYYY HH:MM:SS.TTT)
	PCPCD	(Entry/exit)	I2	Product confidence data (Processing quality)
	PCLAT	(Entry/exit)	I4	Product centre latitude: -90 to +90 (deg*1000)
	PCLON	(Entry/exit)	I4	Product centre longitude: 0 to 360 (deg*1000)
	PCTRK	(Entry/exit)	I4	Satellite track from north: 0 to 360 (deg*1000)
	CLAT	(Entry/exit)	I4	cell latitude: -90 to +90 (deg*1000) (same as PCLAT)
	CLON	(Entry/exit)	I4	cell longitude: 0 to 360 (deg*1000) (same as PCLON)
	INCID	(Entry/Exit)	I2	Incidence angle (deg*1000)
	CLUTTER	(Entry/exit)	I4	Clutter noise estimate (*1000)
	RANRES	(Entry/exit)	I4	Range resolution (pixel spacing): 0 to 1000 (m*1000)
	AZIRES	(Entry/exit)	I4	Azimuth resolution (pixel spacing): 0 to 1000 (m*1000)
	MSCBN	(Entry/Exit)	I4	Maximum spectral component before normalisation

SPIN	(Entry/exit)	I1	12x12 spectral intensity values
NISRA	(Entry/Exit)	I4	Number of image samples in range and azimuth (NR+NA*2^{16})
AZICCO	(Entry/Exit)	I4	Azimuth clutter cut-off
GT731M	(Entry/Exit)	5I4	Spectrum at wavelengths > 731m (1) Total image spectral energy (2) Mean wavelength (m) (3) Wavelength spread (m) (4) Mean direction (deg) (5) Direction spread (deg)
IERR	(Exit)	I4	Error/status code <0 = end of file (for input) 0 = OK >0 = read or write error
BLANK	(Exit)	L4	.T. if product is blank/invalid

INCLUDES FDPTYPES.FOR FDP product and BUFR data types

FUNCTION Reads or writes a nominal UWA product from/to file. For compatibility with ISS the file must have been previously opened with keywords FORM='UNFORMATTED', RECORDTYPE='VARIABLE', RECL=4096 and CARRIAGECONTROL='LIST' on unit 'UNIT'. The RECL and CARRIAGECONTROL keywords are optional for reading but must be used when opening a file for writing. *** *Note On VAX systems RECL is in 4-byte longwords, so the maximum record length is 16384 bytes****.
The file format is compatible with the ESA URA product specification, though not all the Main or Specific Product Header parameters are given values on output.

INTRINSICS BTEST, IBCLR, IBSET

CALLED FROM B2T_UWA, DEBUFR_UWA, ENBUFR_UWA, T2B_UWA

REFERENCES 1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.
2) UWA processing algorithm specification 2.0. ER-TN-ESA-GS-0342, Issue 1.0, Nov 95.

50.

IDENTIFIER	UWI_CF			
TYPE	SUBROUTINE (B0704 in CONVFMT.FOR) - (VAXFORT)			
PURPOSE	Convert UWI parameters between ESA and BUFR formats			
ARGUMENTS	SATID	(Entry/Exit)	I4	Satellite identifier (1 or 2 for ERS-1/2)
	SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
	ORGID	(Entry/Exit)	I4	Originating centre Identifier (253 for ESRIN)
	STNID	(Entry/Exit)	I4	Station identifier (e.g. 13 for Tromso)
	PCTRK	(Entry/Exit)	I4	Satellite track from north, 0-360 (deg*1000)
	DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
	SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
	DTC	(Entry/Exit)	C24	Product date and time string (DD-MMM-YYYY HH:MM:SS.TTT)
	CLAT	(Entry/Exit)	I4	361 cell latitudes (deg*1000)
	CLON	(Entry/Exit)	I4	361 cell longitudes (deg*1000)
	S0	(Entry/Exit)	I4	3*361 cell sigma-zeros (db*10**7)
	AI	(Entry/Exit)	I2	3*361 cell incidence angles (deg*10)
	BA	(Entry/Exit)	I2	3*361 cell beam azimuths (deg*10)
	KP	(Entry/Exit)	I1	3*361 cell kp (percent*10)
	MS	(Entry/Exit)	I1	3*361 cell missing shots
	SPD	(Entry/Exit)	I1	361 cell wind speeds (m/s*20)
	DIR	(Entry/Exit)	I1	361 cell wind directions (deg*2)
	PCD	(Entry/Exit)	I2	361 cell product confidence flags
	CELDAT	(Exit/Entry)	I4	Array of 44*361 BUFR values
	IND	(Entry)	I4	Direction of conversion: >= 1 : to BUFR format <= 0 : to ESA format

INCLUDES	ERRCODES.FOR	Error code definitions
	FDPTYPES.FOR	FDP product and BUFR data types

FUNCTION Converts UWI parameters between ESA format (see Ref 1) and BUFR format (Ref 2). Direction of conversion controlled by value of 'IND'. The BUFR format parameters are contained in the integer array 'CELDAT'.

SUBPROGRAMS HDR_CF

INTRINSICS ABS, BTEST, IAND, IBSET, FLOAT, MAX, MIN, NINT

CALLED FROM B2T_UWI, DEBUFR_UWI, ENBUFR_UWI, T2B_UWI

REFERENCES 1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 Apr 94
2) ERS Products: WMO FM94 BUFR Format. ER-IS-UKM-GS-0001, Issue 4/1, Jan 1999

51.

IDENTIFIER UWI_IO

TYPE SUBROUTINE (U0004 in FDPIO.FOR) - (VAXFORT)

PURPOSE ERS-1/2 UWI product input/output routine

ARGUMENTS

UNIT	(Entry)	I4	File unit number
INPUT	(Entry)	L4	TRUE (.T.) for input, FALSE (.F.) for output
SATID	(Entry/exit)	I4	Satellite identifier (1 for ERS-1, 2 for ERS-2)
SWID	(Entry/Exit)	I4	Software identifier (e.g. 321 for V3.21)
STNID	(Entry/exit)	I4	Station identifier (eg. 1 for Kiruna)
PRDID	(Exit)	I4	Product identifier (8 for UWI)
DTV	(Entry/Exit)	C24	Date and time of ascending node state vector (DD-MMM-YYYY HH:MM:SS.TTT)
SVEC	(Entry/Exit)	I4	6-element ascending node state vector (1)-(3) = x,y,z position (m*100) (4)-(6) = x,y,z velocity (m/s*100000)
DTC	(Entry/exit)	C24	Date and time string (DD-MMM-YYYY HH:MM:SS.TTT)
PCPCD	(Entry/exit)	I2	Product confidence data (Processing quality)
PCLAT	(Entry/exit)	I4	Product centre latitude: -90 to +90 (deg*1000)
PCLON	(Entry/exit)	I4	Product centre longitude: 0 to 360 (deg*1000)
PCTRK	(Entry/exit)	I4	Satellite track from north: 0 to 360 (deg*1000)
CLAT	(Entry/exit)	I4	361 cell latitudes: -90 to +90 (deg*1000)
CLON	(Entry/exit)	I4	361 cell longitudes: 0 to 360 (deg*1000)
S0	(Entry/exit)	I4	3 x 361 sigma-zeros (db*(10**7))
AI	(Entry/exit)	I2	3 x 361 incidence angles (deg*10)
BA	(Entry/exit)	I2	3 x 361 azimuth angles (deg*10)
KP	(Entry/exit)	I1	3 x 361 noise values (percent)
MS	(Entry/exit)	I1	3 x 361 missing counters
SPD	(Entry/exit)	I1	361 wind speeds (m/s*2*10)
DIR	(Entry/exit)	I1	361 wind directions (deg*2)
PCD	(Entry/exit)	I2	361 product confidence data (cell)
IERR	(Exit)	I4	Error/status code <0 = end of file (for input) 0 = OK >0 = read or write error

	BLANK	(Exit)	L4 .T. if product is blank/invalid

INCLUDES FDPTYPES.FOR FDP product and BUFR data types

FUNCTION Reads or writes a nominal UWI product from/to file. For compatibility with ISS the file must have been previously opened with keywords FORM='UNFORMATTED', RECORDTYPE='VARIABLE', RECL=4096 and CARRIAGECONTROL='LIST' on unit 'UNIT'. The RECL and CARRIAGECONTROL keywords are optional for reading but must be used when opening a file for writing. *** *Note On VAX systems RECL is in 4-byte longwords, so the maximum record length is 16384 bytes****.
The file format is compatible with the ESA URA product specification, though not all the Main or Specific Product Header parameters are given values on output.

INTRINSICS BTEST

CALLED FROM B2T_UWI, DEBUFR_UWI, ENBUFR_UWI, T2B_UWI

REFERENCES 1) ERS-2 Kiruna station user interface specification. ER-IS-MDA-GS-2010, Issue 1/0, 5 APR 1994.