

A GUIDE TO THE WMO CODE FORM

FM 94-IX EXT. BUFR

W. Thorpe

Fleet Numerical Oceanography Center

Monterey, California

A GUIDE TO THE WMO CODE FORM FM 94-IX EXT. BUFR

TABLE OF CONTENTS

		Page
TABLE OF CONTENTS		i
LIST OF FIGURES	iii	
LIST OF TABLES		v
INTRODUCTION		vi
 CHAPTER 1. SECTIONS OF A BUFR MESSAGE		 1-1
1.1 Introduction	1-1	
1.2 Specifications of octets within each section	1-1	
1.2.1 Section 0 - Indicator Section	1-3	
1.2.2 Section 1 - Identification Section	1-5	
1.2.3 Section 2 - Optional Section	1-7	
1.2.4 Section 3 - Data Description Section	1-8	
1.2.5 Section 4 - Data Section	1-10	
1.2.6 Section 5 - End Section	1-10	
1.2.7 Required Entries	1-10	
1.2.8 BUFR and Data Management	1-11	
 CHAPTER 2. BUFR TABLES		 2-1
2.1 Introduction	2-1	
2.2 Table A - Data Category	2-1	
2.3 Table B - Classification of Elements	2-2	
2.3.1 Data Replication	2-4	
2.4 Table C - Data Description Operators	2-5	
2.5 Table D - Lists of Common Sequences	2-5	
2.6 Message Layout	2-7	
2.6.1 Comparison of BUFR and Character Code Bit Counts	2-7	
2.7 Code Tables and Flag Tables	2-12	
2.7.1 Code Tables	2-12	
2.7.2 Flag Tables	2-12	
2.7.3 Flags	2-12	
2.8 Local Tables	2-13	
 CHAPTER 3. USING DATA REPLICATION		 3-1
3.1 Introduction	3-1	
3.2 Data Replication Examples	3-1	
 CHAPTER 4. DATA COMPRESSION		 4-1
4.1 Introduction	4-1	

CHAPTER 5. TABLE C - DATA DESCRIPTION OPERATORS	5-1
5.1 Introduction	5-1
5.2 Changing Data Width, Scale and Reference Values	5-1
5.2.1 Changing Reference Value Only	5-7
5.3 Add Associated Field	5-10
5.4 Encoding Character Data	5-15
5.5 Signifying Length of Local Descriptors	5-16
CHAPTER 6. Quirks, Advanced Features, and Special Uses of BUFR	6-1
6.1 Introduction	6-1
6.2 Section 0 - Indicator Section	6-1
6.2.1 Edition Number Changes	6-1
6.2.2 Maximum Size of BUFR Records	6-3
6.3 Section 1 - Identification Section	6-3
6.3.1 Master Tables, Version Numbers, and Local Tables	6-3
6.3.2 Originating Centre (or Center)	6-4
6.3.3 Update Sequence Number	6-5
6.3.4 Optional Section 2	6-5
6.3.5 BUFR Message Sub-Type	6-5
6.3.6 Date/Time	6-7
6.3.7 "Reserved for use..."	6-7
6.4 Section 2 - Optional Section - Example of Data Base Keys	6-8
6.4.1 U.S. National Meteorological Center Usage	6-8
6.4.1.1 BUFR as a Data Base Storage Format	6-9
6.5 Section 3 - Data Description Section	6-10
6.5.1 Data Subsets	6-10
6.5.2 Observed or "other data"	6-11
6.5.3 Data Descriptors	6-11
6.5.3.1 Descriptors for "Coordinates"	6-12
6.5.3.2 Replication, Increments and "Run-Length Encoding"	6-14
6.5.3.3 The Associated Field	6-17
6.5.3.4 Changing Descriptors "On the Fly"	6-17
6.5.3.5 BUFR Records in Archives	6-18
CHAPTER 7. Use of Binary Representation at ECMWF	7-1
7.1 Introduction	7-1
7.2 Operational Data Management	7-1
7.3 Use of BUFR	7-2
7.4 Use of GRIB	7-3
7.5 Concluding Remarks	7-3

LIST OF FIGURES

Figure

1-1	Example of a complete BUFR message containing 52 octets	1-2
1-2	Section 0	1-4
1-3	Section 1	1-6
1-4	Section 3	1-9
1-5	Section 4	1-12
1-6	Section 4 data as described by descriptors	1-12
1-7	Section 5	1-12
1-8	Required entries in sample BUFR message	1-13
2-1	Example of surface observations sequence using Table D descriptor 3 07 002	2-9
2-2	BUFR message of 1 surface observation using Table D descriptor 3 07 002	2-10
2-3	BUFR message of 448 surface observations using Table D descriptor 3 07 002	2-11
2-4	Table reservations	2-13
2-5	Example of surface observations sequence using Table D descriptor 3 07 002 and a local descriptor	2-15
2-6	BUFR message of 443 surface observations using 2 descriptors	2-16
3-1	Example of TEMP observations sequence using delayed replication	3-3
4-1	Comparison of non-compressed and compressed data in Section 4	4-3
4-2	BUFR message of 6 subsets in non-compressed	4-7

	form	
4-3	BUFR message of 6 subsets in compressed form	4-8
4-4	BUFR message of 1898 subsets in non-compressed form	4-9
4-5	BUFR message of 4267 subsets in compressed form	4-10
5-1	Change reference value of geopotential	5-9
5-2	Example of TEMP observations sequence using delayed replication and quality control information	5-14
5-3	Example of surface observations with local descriptor and data descriptor 2 06 Y	5-17

LIST OF TABLES

Table

2-1	BUFR Table A - Data Category	2-1
2-2	BUFR Table D - List of Common Sequences	2-6
5-1	BUFR Table C - Data Description operators	5-2

INTRODUCTION

The World Meteorological Organization (WMO) code form FM 94-IX Ext. BUFR(Binary Universal Form for the Representation of meteorological data) is a binary code designed to represent, employing a continuous binary stream, any meteorological data. There is, however, nothing uniquely meteorological about BUFR. The meteorological emphasis is the result of the origin of the code. The code form may be applied to any numerical or qualitative data type.

BUFR is the result of a series of informal and formal "expert meetings" and periods of experimental usage by several meteorological data processing centers. The WMO Commission for Basic Systems (CBS) approved BUFR at its January/February 1988 meeting. Changes were introduced at the CBS Working Group on Data Management, Sub-Group on Data Representation meetings in May, 1989 and October 1990. The changes introduced at the October 1990 meeting were of such magnitude that BUFR, Edition 2 was defined, with an effective date of November 7, 1991.

The key to understanding the power of BUFR is the code's self-descriptive nature. A BUFR "message" (or record, the terms are interchangeable in this context) containing observational data of any sort also contains a complete description of what those data are: the description includes identifying the parameter in question, (height, temperature, pressure, latitude, date and time, whatever), the units, any decimal scaling that may have been employed to change the precision from that of the original units, data compression that may have been applied for efficiency, and the number of binary bits used to contain the numeric value of the observation. This data description is all contained in tables which are the major part of the BUFR documentation.

The strength of this self-descriptive feature is in accommodating change. For example, if new observations or observational platforms are developed, there is no need to invent a new code form to represent and transmit the new data; all that is necessary is the publication of additional data description tables. Similarly for the deletion of possibly outdated observations: instead of having to send "missing" indicators for a long period while awaiting a change to a fixed format code, the "missing" data are simply not sent in the message and the data description section is adjusted accordingly. The data description tables are not changed, however, so that archives of old data may be retrieved.

This self-descriptive feature leads to another advantage over character oriented codes - The relative ease of decoding a BUFR message. Where a large number of specialized and complex programs are now needed to decode the plethora of character codes in current use, it is entirely feasible to write a single "universal BUFR decoder" program capable of decoding any BUFR message. It is not a trivial task to write such a BUFR decoder, but once it is done, it is done for all time. The program will not have to change with changes in observational practices; only the tables will need to be augmented, a relatively trivial task.

The development of BUFR has been synonymous with the development of the data description language that is integral to it. Indeed the major portion of the full description of BUFR is a description of the vocabulary and syntax of the data description language. The definition of the data description language, and the "descriptors" that are its vocabulary, are what give BUFR its "universal" aspect: any piece of information can be described in the language, not just meteorological observations.

The other major aspect of BUFR is reflected in the first initial, "B"; BUFR is a purely binary or bit oriented form, thus making it both machine dependent and, at the same time, machine independent. The dependency comes in the construction or interpretation of BUFR messages: there is not much for a human to look at (unless she is very patient) as all the numbers in a message, whether data descriptors or the data themselves, are binary integers. And that, of course, leads to the machine independence: with BUFR consisting entirely of binary integers any brand of machine can handle BUFR as well as any other.

The binary nature of BUFR leads to another advantage over character codes: the ease and speed of converting the message into an internally useful numeric format. With character codes the conversion from ASCII (or EBCDIC) to integer or floating point is expensive relative to the conversion from binary integers to floating point. The latter is all that BUFR requires. In some tests, the European Centre for Medium-Range Weather Forecasts found a speedup of better than 6 times in decoding BUFR messages over the corresponding TEMP (WMO Radiosonde character code FM 35-IX Ext.) messages. The BUFR data also required about half the machine memory as the character data.

All of this does assume the availability of well designed computer programs that are capable of parsing the descriptors, which can be a complex task, matching them to the bit stream of data and extracting the numbers from the stream, responding properly to the arrival of new (or the departure of old) data descriptors, and reformatting the numbers in a way suitable for subsequent calculations. The bit oriented nature of the message also requires the availability of bit transparent communications systems such as the x.25 protocol. Such protocols have various error detecting schemes built in so there need be little concern about the corruption of information in the transmission process.

Dr. John D. Stackpole
NOAA/NWS
National Meteorological Center
Camp Springs, MD 20746 U.S.A
CHAPTER 1

Sections of a BUFR Message

1.1 Introduction. The term "message" refers to BUFR being used as a data transmission format; however, BUFR can, and is, used in several meteorological data

processing centers as an on-line storage format as well as a data archiving format.

1.2 Specifications of Octets Within Each Section. For transmission of data, each BUFR message consists of a continuous binary stream comprising 6 sections.

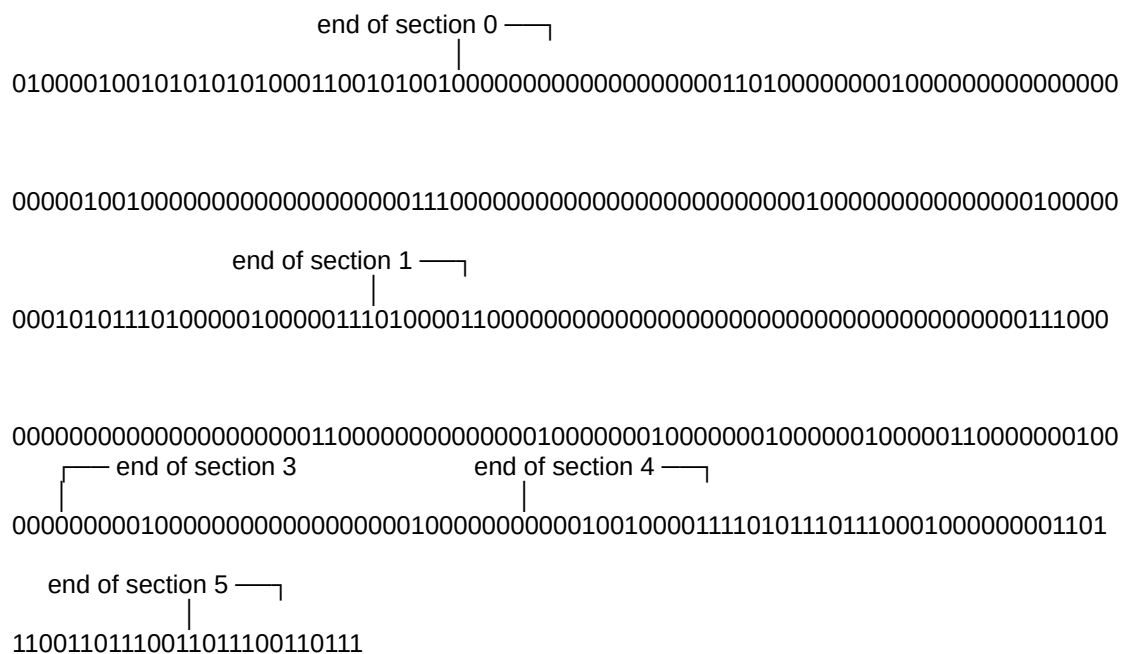
CONTINUOUS BINARY STREAM					
section	section	section	section	section	section
0	1	2	3	4	5

Section number	Name	Contents
0	indicator section	"BUFR" (coded according to the CCITT International Alphabet No. 5, which is functionally equivalent to ASCII), length of message, BUFR edition number
1	identification	length of section, identification of the section message
2	optional section	length of section and any additional items for local use by data processing centers
3	data description	length of section, number of data subsets, data category flag, data compression flag, and a collection of data descriptors which define the form and content of individual data elements
4	data section	length of section and binary data
5	end section	"7777" (coded in CCITT International Alphabet No. 5)

Each of the sections of a BUFR message is made up of a series of octets. The term octet, meaning 8 bits, was coined to avoid having to continually qualify byte as an 8-bit byte. Also, in French, the words "byte" and "bit" are pronounced the same (as "beet"), "octet" clearly avoids that problem, too. An individual section shall always consist of an even number of octets, with extra bits added on and set to zero when necessary. Within each section, octets are numbered 1, 2, 3, etc., starting at the beginning of each section. Bit positions within octets are referred to as bit 1 to bit 8, where bit 1 is the most significant, leftmost, or high order bit. An octet with only bit 8 set would have the integer value 1.

Theoretically there is no upper limit to the size of a BUFR message but, by convention, BUFR messages are restricted to 15000 octets or 120000 bits. This limit is to allow an

entire BUFR message to be contained within memory of most computers for decoding. It is also a limit set by the capabilities of the Global Telecommunications System (GTS) of the WMO. The BLOK feature, described elsewhere, can be used to break very long BUFR messages into parts, if necessary.



1.2.1 Section 0 - Indicator section.

Octet No. contents

1 - 4 "BUFR" (coded according to the CCITT International
 Alphabet No. 5)

5 - 7 Total length of BUFR message, in octets (including
 Section 0)

8 BUFR edition number (currently 2)

The earlier editions of BUFR did not include the total message length in octets 5-7. Thus, in decoding BUFR Edition 0 and 1 messages, there was no way of determining the entire length of the message without scanning ahead to find the individual lengths of each of the sections. Edition 2 eliminates this problem by including the total message length right up front. By design, in BUFR Edition 2, octet 8, containing the BUFR Edition number, is in the same octet position relative to the start of the message as it was in Editions 0 and 1. By keeping the relative position fixed, a decoder program can determine, at the outset, which BUFR version was used for a particular message and then behave accordingly. This means, for example, that archives of old (pre-Edition 2) records need not be updated.

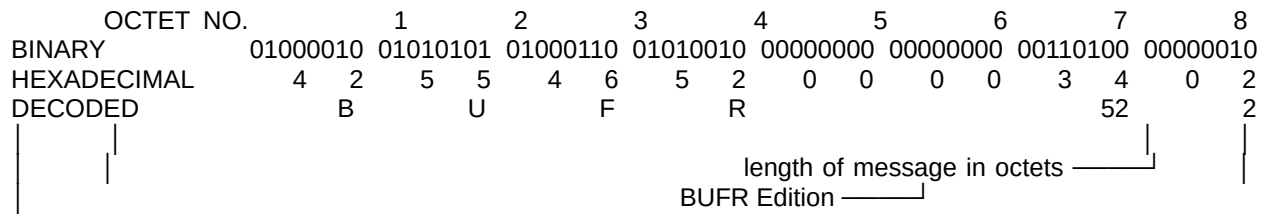


Figure 1-2. Section 0

1.2.2 Section 1 - Identification Section.

CONTINUOUS BINARY STREAM					
section	SECTION	section	section	section	section
0	1	2	3	4	5

Octet No. contents

1 - 3 Length of section, in octets

4 BUFR master table (zero if standard WMO FM 94-IX EXT.
 BUFR tables are used - provides for BUFR to be used to represent
 data from other disciplines, and with their own versions of master tables
 and local tables)

5 - 6 Originating centre: code table 0 01 031

- 7 Update sequence number (zero for original BUFR messages; incremented for updates)
- 8 Bit 1 = 0 No optional section
= 1 Optional section included
- Bits 2 - 8 set to zero (reserved)
- 9 Data Category type (BUFR Table A)
- 10 Data Category sub-type (defined by local ADP centres)
- 11 Version number of master tables used (currently 2 for WMO FM 94-IX EXT. BUFR tables)
- 12 Version number of local tables used to augment the master table in use
- 13 Year of century
- 14 Month
- 15 Day
- 16 Hour
- 17 Minute
- 18 - Reserved for local use by ADP centres

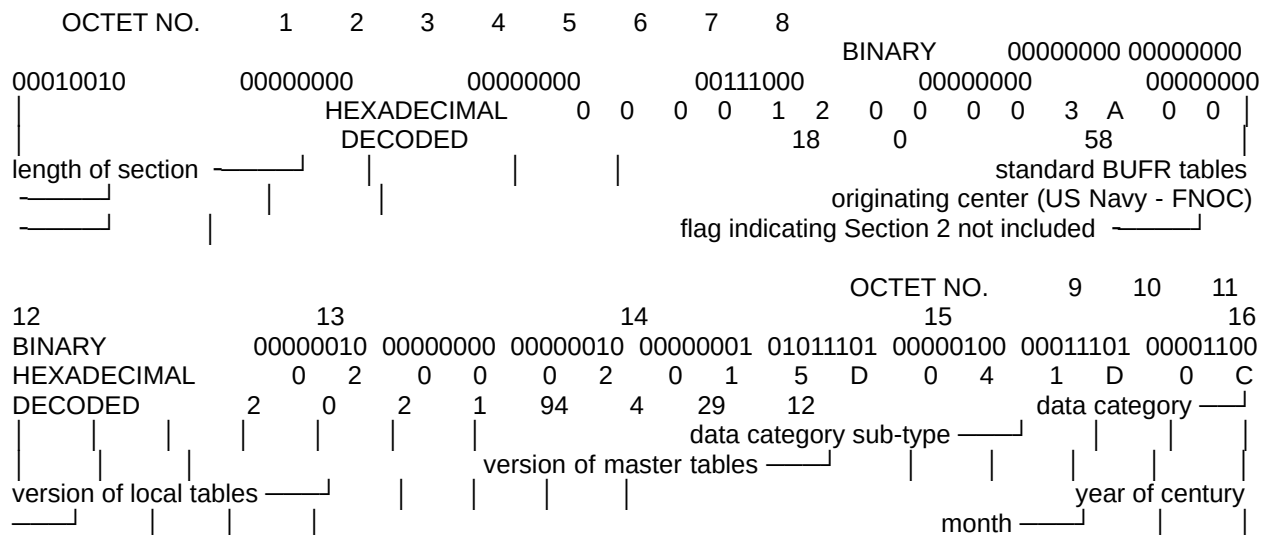




Figure 1-3. Section 1

The length of section 1 can vary between BUFR messages. Beginning with Octet 18, a data processing center may add any type of information as they choose. A decoding program may not know what that information may be. Knowing what the length of the section is, as indicated in octets 1-3, a decoder program can skip over the information that begins at octet 18 and position itself at the next section, either section 2, if included, or section 3. Bit 1 of octet 8 indicates if section 2 is included. If there is no information beginning at octet 18, one octet must still be included (set to 0) in order to have an even number of octets within the section.

1.2.3 Section 2 - Optional Section.

CONTINUOUS BINARY STREAM					
section	section	SECTION	section	section	section
0	1	2	3	4	5

Octet No.	Contents
1 - 3	Length of section, in octets
4	set to zero (reserved)
5 -	Reserved for use by ADP centres

Section 2 may or may not be included in any BUFR message. When it is contained within a BUFR message, bit 1 of octet 8, Section 1, is set to 1. If Section 2 is not included in a message then bit 1 of octet 8, Section 1 is set to 0. Section 2 may be used for any purpose by an originating center. The only restrictions on the use of Section 2 are that octets 1 - 3 are set to the length of the section, octet 4 is set to zero and the total length of the section contains an even number of octets.

A typical use of this optional section could be in a data base context. The section might contain pointers into the data section of the message, pointers which indicate the relative location of the start of individual sets of observations (one station's worth, for example) in the data. There could also be some sort of index term included, such as the WMO block and station number. This would make it quite easy to find a particular observation quickly and avoid decoding the whole message just to find one or two specific data elements.

1.2.4 Section 3 - Data description section.

CONTINUOUS BINARY STREAM					
section	section	section	SECTION	section	section
0	1	2	3	4	5

Octet No.	Contents
1 - 3	Length of section, in octets
4	set to zero (reserved)
5 - 6	number of data subsets
7	Bit 1 = 1 observed data = 0 other data Bit 2 = 1 compressed data = 0 non-compressed data Bit 3 - 8 set to zero (reserved)
8 -	A collection of descriptors which define the form and content of individual data elements comprising one data subset in the data section.

If octets 5-6 indicate that there is more than one data subset in the message, with the total number of the subsets given in those octets, then multiple sets of observations, all with the same format (as described by the data descriptors) will be found in Section 4. This is, for example, a means of building "collectives" of observations. Doing so realizes a large portion of the potential of efficiency in BUFR.

In the flag bits of octet 7, "observed data" is taken to mean just that; "other data", is by custom, if not explicit statement, presumed to be forecast information, or possibly some form of "observation", indirectly derived from "true" observations. The nature of "data compression" will be described in Chapter 4.

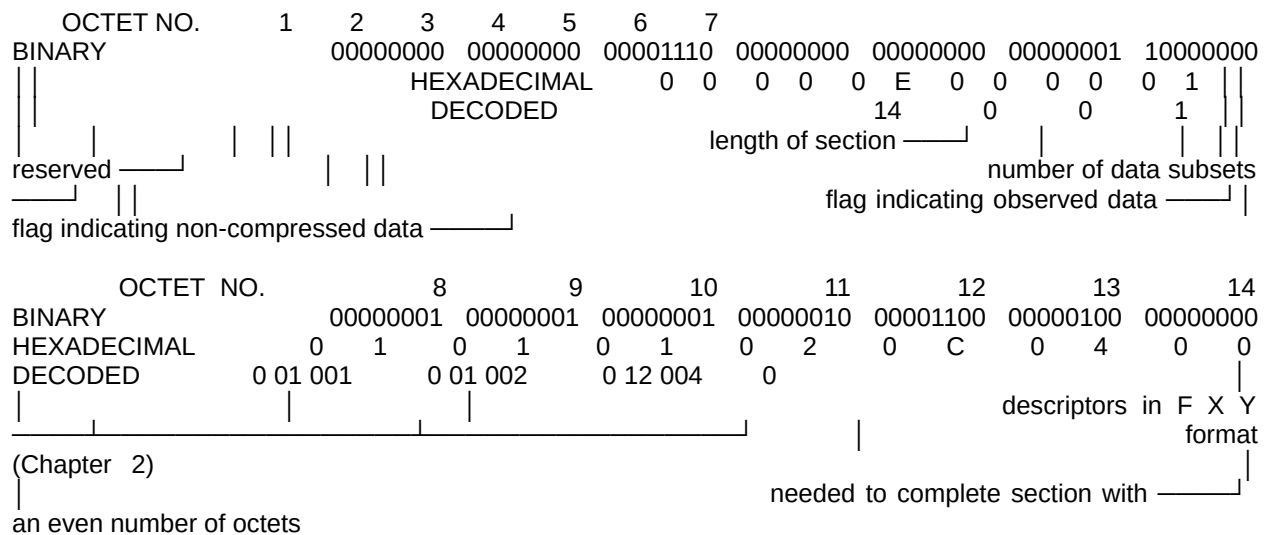


Figure 1-4. Section 3

1.2.5 Section 4 - Data Section.

CONTINUOUS BINARY STREAM					
section	section	section	section	SECTION	section
0	1	2	3	4	5

Octet No.	Contents
1 - 3	Length of section, in octets
4	set to zero (reserved)
5	Binary data as defined by descriptors which begin at octet 8, Section 3.

1.2.6 Section 5 - End Section.

CONTINUOUS BINARY STREAM

Section 5 - octets 1 - 4

Figure 1-8 is the same BUFR message as in Figures 1-1 to 1-7. The shaded areas in Figure 1-8 are those octets which are required in any BUFR message. Not included in the shaded areas are descriptors contained in octets 8 - 14 of Section 3 and the data in Octets 5 - 8 of section 4.

1.2.8 BUFR and Data Management. Sections 3 and 4 of BUFR contain all of the information necessary for defining and representing data. The remaining sections are defined and included purely as aids to data management. Key information within these sections is available from fixed locations relative to the start of each section. It is thus possible to categorize and classify the main attributes of BUFR data without decoding the data description in Section 3, and the data in Section 4.

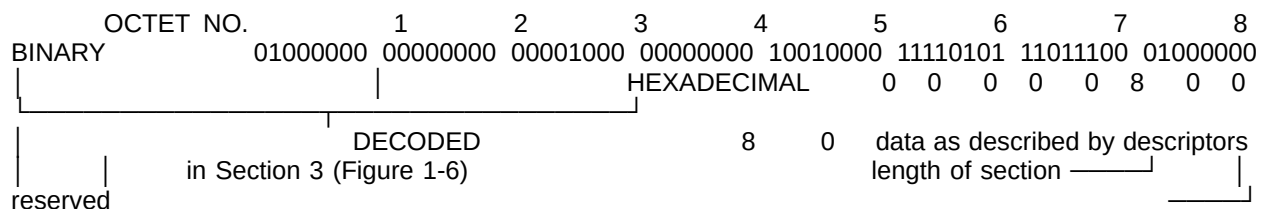


Figure 1-5. Section 4

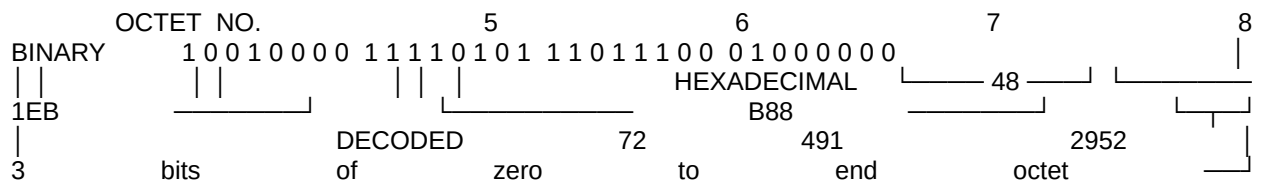


Figure 1-6. Section 4 data as described by descriptors

CHAPTER 2

BUFR Tables

2.1 Introduction. BUFR employs 3 types of tables: BUFR tables, code tables and flag tables.

The tables in BUFR that contain information to describe, classify and define the contents of a BUFR message are called BUFR tables. There are 4 tables defined: Tables A, B, C and D.

2.2 TABLE A - Data Category. Table A is referred to in Section 1 and provides a quick check for the type of data represented in the message. Of the 256 possible entries for Table A, 17 are currently defined:

Table 2-1. BUFR TABLE A - DATA CATEGORY

Code Figure	Meaning
0	Surface data - land
1	Surface data - sea
2	Vertical soundings (other than satellite)
3	Vertical soundings (satellite)
4	Single level upper-air data (other than satellite)
5	Single level upper-air data (satellite)
6	Radar data
7	Synoptic data
8	Physical/chemical constituents
9	Dispersal and transport
10	Radiological data
11	BUFR tables, complete replacement or update
12	Surface data (satellite)
13-19	Reserved
20	Status information
21	Radiances
22-30	Reserved
31	Oceanographic data
32-100	Reserved
101	Image data
102-255	Reserved

The setting of one of the code figures for Table A (Table 2-1) in octet 9 of Section 1 is actually redundant. The descriptors used in Section 3 of a message define the data in Section 4, regardless of the Table A code figure. Decoding programs may well reference Table A, finding it useful to have a general classification of the data available

prior to actually decoding the information and passing it on to some subsequent application program.

2.3 TABLE B - Classification of Elements. Table B is referenced in Section 3 of a BUFR message and contains descriptions of parameters encoded in Section 4. Table B entries, as described in the WMO Manual On Codes, Volume 1, Part B, consist of 6 entities:

a descriptor consisting of the 3 parts F X and Y

element name

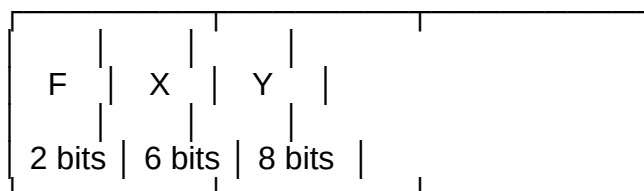
units: basic (SI) units for the element

scale: factor (equal to 10 to the power [scale]) by which the element has been multiplied prior to encoding

reference value: a number to be subtracted from the element, after scaling, (if any), and prior to encoding

data width, in bits, the element requires for representation in Section 4

A Table B descriptor consists of 16 bits (2 octets) divided into 3 parts, F, X and Y.



F (2 bits) indicates the type of descriptor. In 2 bits there are 4 possibilities, 0, 1, 2 and 3. The numeric value of the 2 bit quantity F, indicates the type of descriptor.

- F = 0 Element descriptor (Table B entry)
- F = 1 Replication operator
- F = 2 Operator descriptor (Table C entry)
- F = 3 Sequence descriptor (Table D entry)

X (6 bits) indicates the class or category of descriptor. There are 64 possibilities, classes 00 to 63. Thus far, 28 classes have been defined.

Y (8 bits) indicates the entry within an X class. 8 bits will yield 256 possibilities within

each of the 64 classes. There are a varying number of entries within each of the 28 classes that are currently defined.

It is the F X Y descriptors in Section 3 that refer to data represented in Section 4. The 16 bits of F X and Y are not to be treated as a 16 bit numeric value, but rather as 16 bits divided into 3 parts, where each part (F, X and Y) are in themselves 2, 6 and 8 bit numeric values. Some examples of descriptors with their corresponding bit settings:

Descriptor	F	X	Y
0 01 001	00	000001	00000001 (Figure 1-4)
1 02 006	01	000010	00000110
2 01 131	10	000001	10000011
3 07 002	11	000111	00000010

If the following descriptors were contained in Section 3:

0 01 001 0 01 002 0 02 001 0 04 001 0 04 002
0 04 003 0 04 004 0 04 005 0 05 002 0 06 002

these descriptors would refer to the following extracts from BUFR Table B:

Table Reference	Element Name	Units	Scale	Reference Value	Data (Bits)	Width
F X Y						
0 01 001	WMO block number	numeric	0	0	0	7
0 01 002	WMO station number	numeric	0	0	0	10
0 02 001	Type of station	code table	0	0	0	2
0 04 001	Year	Year	0	0	0	12
0 04 002	Month	Month	0	0	0	4
0 04 003	Day	Day	0	0	0	6
0 04 004	Hour	Hour	0	0	0	5
0 04 005	Minute	Minute	0	0	0	6
0 05 002	Latitude	Degree	2	-9000	0	15
	(coarse accuracy)					
0 06 002	Longitude	Degree	2	-18000	0	16
	(coarse accuracy)					

The element name is a plain language description of the element entry of the table.

The units of Table B entries refer to the format of how the data in Section 4 is represented. The data may be numeric as in the case of a WMO block number, character data as in the case of an aircraft identifier. When data is in character form, the character representation is always according to the CCITT International Alphabet No. 5. The units may also refer to a code or flag table, where the code or flag table is described in the WMO Manual On Codes using as the code or flag table number the

same number as the F X Y descriptor. Other units are in Standard International (SI) units, such as meters or degrees Kelvin.

The scale refers to the power of 10 that the element in Section 4 has been multiplied by in order to retain the desired precision in the transmitted data. For example, the units of latitude are whole degrees in Table B. But this is not precise enough for most usages, therefore the elements are to be multiplied by 100 (10^2) so that the transmitted precision will be centidegrees, a more useful precision. On the other hand, the (SI) unit of pressure in Table B is Pascals, a rather small unit that would result in unnecessarily precise numbers being transmitted. The BUFR Table B calls for pressure to be divided by 10 (10^{-1}) resulting in a transmitted unit of 10ths of hPa, or tenths of millibars, a more reasonable precision for meteorological usage. These precisions can be changed on the fly, so to speak, if the table values are not appropriate in special cases. This is done through the use of "operator descriptors" - see below, 2.4 Table C.

The reference value is a value that is to be subtracted from the data after multiplication by the scale factor, if any, before encoding into Section 4 in order to produce, in all cases, a positive value. In the case of latitude and longitude, south latitude and west longitude are negative before applying the reference value. If, for example, a position of 35.50 degrees south latitude were being encoded, multiplying -35.50 by 100 (scale of 2) would produce -3550. Subtracting the reference value -9000 would give 5450 that would be encoded in Section 4. To obtain the original value in decoding Section 4, adding back the -9000 reference value to 5450 would result in -3550, then dividing by the scale (100) would obtain -35.50.

The data width of Table B entries is a count of how many bits the largest possible value of an individual data item of Section 4 occupies.

In those instances where a Table B descriptor defines an element of data in Section 4, where that element is missing for a given subset, then all bits for that element will be set to 1's in Section 4.

Obviously, without an up-to-date Table B, a decoder program would not be able to determine the form or content of data appearing in Section 4.

2.3.1 Data Replication. A special descriptor called the replication operator ($F = 1$) is used to define a range of subsequent descriptors, together with a replication factor. This enables the appropriate descriptors to be considered to be repeated a number of times. In general for data replication, X indicates the number of immediately following descriptors that are to be replicated as a repeated set, and Y indicates the total number of replications. This, of course, implies, that the same pattern will be found in Section 4, the data section. This ability to describe a repeated pattern in the data by a single set of descriptors contributes to the efficiency of BUFR.

As an example, consider the following sequence appears in Section 3:

1 02 006 0 07 004 0 01 003

the meaning of 1 02 006 is that the next 2 descriptors are repeated 6 times, or the equivalent set of descriptors:

0 07 004 0 01 003 0 07 004 0 01 003 0 07 004 0 01 003
0 07 004 0 01 003 0 07 004 0 01 003 0 07 004 0 01 003

A special form of the replication operator allows the replication factor to be stored with the data in Section 4, rather than with the descriptor in Section 3. This special form is called delayed replication. It is indicated by Y = 0. It allows the data to be described in a general way, with the number of replications being different from subset to subset. Since the data now contains an additional data element, the actual replication count, a descriptor must be added to Section 3 to account for, and describe, this (special) data element. The appropriate descriptor is found in Class 31. Special note: the 0 31 YYY (delayed replication factor) descriptor follows immediately after the 1 X 000 (delayed replication) descriptor but is NOT included in the count (X) of the following descriptors to be replicated.

Another form of delayed replication enables both the data description and the corresponding data item or items to be repeated. Entries in Class 31 of Table B are used in association with the delayed replication operator to enable this to be done.

2.4 Table C - Data Description Operators. Table C data description operators (Chapter 5) are used when there is a need to redefine Table B attributes temporarily, such as the need to change data width, scale or reference value of a Table B entry. Table C is also used to add associated fields such as quality control information, indicate characters as data items, and signify data width of local descriptors.

2.5 Table D - Lists of Common Sequences. Table D contains descriptors which describe additional descriptors. A single descriptor used in Section 3 with F = 3 is a pointer to a Table D entry which contains other descriptors. If the Table D descriptor 3 01 001 were used in Section 3, the expansion of that descriptor is two Table B descriptors, 0 01 001 and 0 01 002.

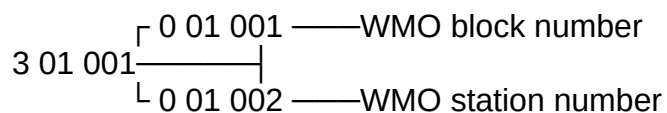
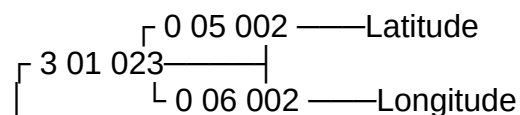
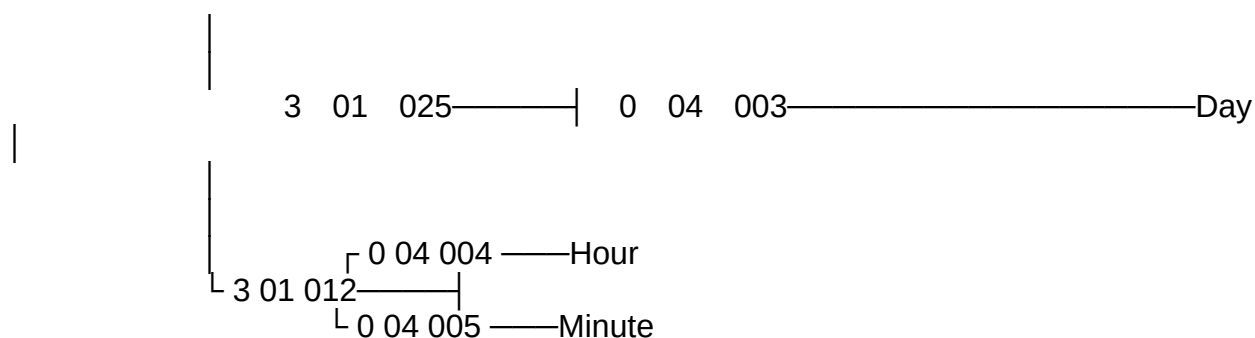


Table D descriptors may also refer to an expansion list of descriptors that contain additional Table D descriptors. The descriptor 3 01 025 expands to 3 01 023, 0 04 003 and 3 01 012. In the expansion, 3 01 023 additionally expands to 0 05 002 and 0 06 002. The remaining descriptor 3 01 012 expands to 0 04 004 and 0 04 005. Thus, the single Table D descriptor 3 01 025 expands to a total of 5 separate Table B entries.





The order of the data in Section 4 is then according to the following sequence of Table B entries: 0 05 002 0 06 002 0 04 003 0 04 004 0 04 005.

There are currently defined 19 categories of common sequences in Table D (Table 2-2).

Table 2-2. BUFR Table D list of common sequences

F	X	CATEGORY OF SEQUENCES
3	00	BUFR table entries sequences
3	01	Location and identification sequences
3	02	Meteorological sequences common to surface data
3	03	Meteorological sequences common to vertical sounding data
3	04	Meteorological sequences common to satellite observations
3	05	Reserved
3	06	Meteorological or oceanographic sequences common to oceanographic observations
3	07	Surface report sequences (land)
3	08	Surface report sequences (sea)
3	09	Vertical sounding sequences (conventional data)
3	10	Vertical sounding sequences (satellite data)
3	11	Single level report sequences (conventional data)
3	12	Single level report sequences (satellite data)
3	13	Sequences common to image data
3	14	Reserved
3	15	Oceanographic report sequences
3	16	Synoptic feature sequences
3	18	Radiological report sequences
3	21	Radar report sequences

Any BUFR message may be encoded without using Table D. The data description contained within Section 3 can be accomplished entirely by using only element descriptors of Table B and operator descriptors of Table C. To do so, however would involve considerable overhead in terms of the length of the Section 3 data description.

The use of Table D is another major contributor to the efficiency of BUFR.

2.6 Message Layout. Figure 2-1 illustrates how the single descriptor 3 07 002 expands into 2 more Table D descriptors, 3 01 032 and 3 02 011. The descriptor 3 01 032 further expands into 5 more descriptors 3 01 001, 0 02 001, 3 01 011, 3 01 012 and 3 01 024. As is shown in Figure 2-1, descriptors in Table D may themselves refer to Table D, provided no circularity results on repeated expansion. Completion of the expansion process leads to a total of 31 Table B descriptors. The 16 bits in Section 3 taken by the descriptor 3 07 002 results in a savings of 480 bits (30 x 16 bits) over what the 31 Table B descriptors would occupy in bits.

Table D has been limited to lists of descriptors likely to be most frequently used. Table D was not designed to be comprehensive of all sequences likely to be encountered. To do so would require an excessively large Table D and would reduce considerably flexibility when encoding minor differences in reporting practices. More flexibility is retained if the Data Description Section contains several descriptors.

A complete layout of a BUFR message containing just 1 surface observation is illustrated in Figure 2-2. As indicated in octets 5-7 of Section 1, there are a total of 78 octets in the message, or 624 bits. Of the 624 bits, 267 are for the actual parameters of data (Figure 2-1) and the remaining 357 bits are BUFR overhead. BUFR overhead in this context is the number of bits that are not actual surface data. In this example there are more bits used for the overhead than for the surface data.

Figure 2-3 is a complete layout of a BUFR message containing the maximum number of 448 subsets to fit within the 15000 octet limit. This message would contain 14996 octets or 119968 bits. Of these 119968 bits, 119616 are data and 352 bits are BUFR overhead. The 5 bit difference in overhead from Figure 2-2 (357 bits) and Figure 2-3 (352 bits) is due to the number of bits set to 0 at the end of Section 4 in order to complete the section at the end of an even numbered octet. For 1 subset of 267 bits, 5 additional bits are needed to complete the octet. For 448 subsets, or 119616 bits, no additional bits are needed to complete the last octet.

2.6.1 Comparison of BUFR and Character Code Bit Counts. The surface observations illustrated in Figures 2-1 to 2-3 are the equivalent of the following parameters in the WMO code form FM 12-IX Ext. SYNOP:

```
YYGGiw llll iRixhVV Nddff 1snTTT 2snTdTdTd 3PoPoPoPo  
4PPPP 5appp 7wwW1W2 8NhCLCMCH
```

Data encoded in this form would consist of 55 characters plus 10 spaces between each group of 5 characters for a total of 65 characters. For transmission purposes these 65 characters would require a total number of 520 bits (65 X 8 bits per character). A complete BUFR message with 1 observation (Figure 2-2) requires 78 octets or 624 bits, 104 more than the corresponding character representation. Of these 624 bits, 267 are

taken by the surface observation and 357 as BUFR overhead. If, however, 448 observations in character form were transmitted, the total number of bits would be 232960 (520 X 448). The corresponding BUFR representation (Figure 2-3) would require 14996 octets, or 119968 bits, a savings of 112992 bits over the character representation. The 112992 bits is equivalent to 217 observations in character form or 423 observations in BUFR, not counting the BUFR overhead. While these numbers may be viewed in different ways, the real significance is that BUFR is far more efficient, in terms of number of bits to represent a meteorological observation, than character forms.

SECTION 4
WIDTH IN BITS

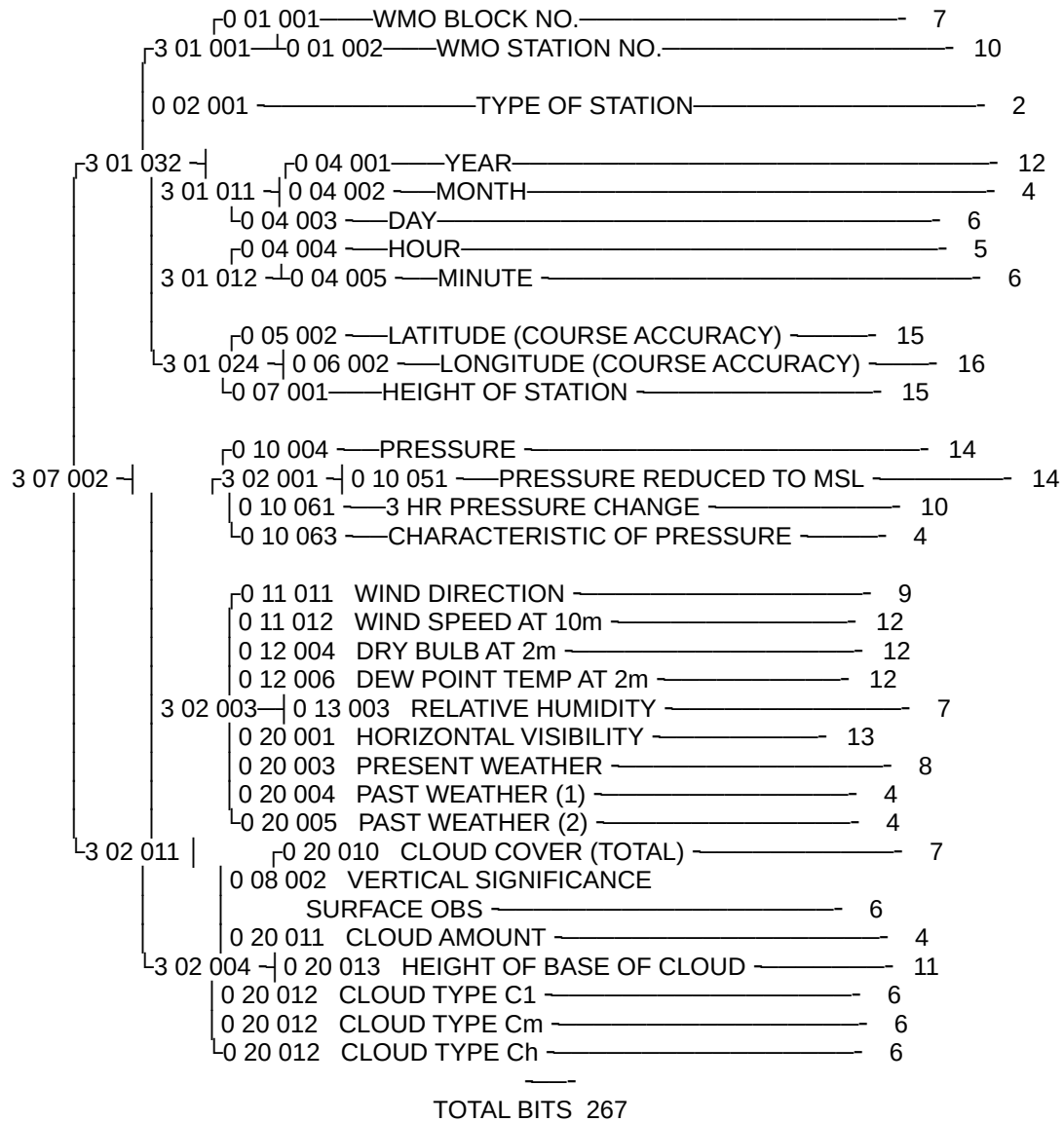


Figure 2-1. Example of surface observations sequence using
Table D descriptor 3 07 002

Section	Octet in	Encoded	Description
Octet No.	Message	Value	
Section 0			
(indicator	1-4	1-4	BUFR encoded international CCITT section)
	5-7	5-7	Alphabet No. 5
	8	8	78 total length of message (octets)
			2 BUFR edition number
Section 1			
(identification	1-3	9-11	length of section (octets)
section)	4	12	0 BUFR master table
	5-6	13-14	58 originating center (U.S. Navy - FNOG)
	7	15	0 update sequence number
	8	16	0 indicator that Section 2 not included
	9	17	0 Table A - surface land data
	10	18	0 BUFR message sub-type
	11	19	2 version number of master tables
	12	20	0 version number of local tables
	13	21	92 year of century
	14	22	4 month
	15	23	18 day
	16	24	0 hour
	17	25	0 minute
	18	26	0 reserved for local use by ADP centers (also needed to complete even number of octets for section
Section 3			
(Data	1-3	27-29	10 length of section (octets)
description	4	30	0 reserved
section)	5-6	31-32	1 number of data subsets
	7	33	bit 1=1 flag indicating observed data
	8-9	34-35	3 07 002 Table D descriptor for surface land in F X Y format
	10	36	0 need to complete section with an even number of octets
Section 4			
(Data	1-3	37-39	38 length of section (octets)
section)	4	40	0 reserved
	5-38	41-74	data continuous bit stream of data for 1 observations, 267 bits plus 5 bits to end on even octet (see Figure 2-1 for expansion)
Section 5			
(End section)	1-4	75-78	7777 encoded CCITT International Alphabet No. 5

Figure 2-2. BUFR message of 1 surface observation using Table D descriptor 3 07 002

Section	Octet in	Encoded	Description
Octet No.	Message	Value	
Section 0 (indicator)	1-4	1-4	BUFR encoded international CCITT section) Alphabet No. 5
	5-7	5-7	14996 total length of message (octets)
	8	8	2 BUFR edition number Section 1
(identification section)	1-3 4	9-11 12	18 length of section (octets)
	5-6	13-14	58 BUFR master table
	7	15	0 originating center (U.S. Navy - FNOC)
	8	16	0 update sequence number
			0 indicator that Section 2 not included
	9	17	0 Table A - surface land data
	10	18	0 BUFR message sub-type
	11	19	2 version number of master table
	12	20	0 version number of local tables
	13	21	92 year of century
	14	22	4 month
	15	23	18 day
	16	24	0 hour
	17	25	0 minute
	18	26	0 reserved for local use by ADP centers (also needed to complete even number of octets for section Section 3
(Data description section)	1-3 4	27-29 30	10 length of section (octets)
	5-6	31-32	448 reserved
	7	33	0 number of data subsets
	8-9	34-35	3 07 002 bit 1=1 flag indicating observed data
	10	36	0 Table D descriptor for surface land in F X Y format
			0 need to complete section with an even number of octets
Section 4 (Data section)	1-3 4	37-39 40	14956 length of section (octets)
	5-14956	41-14992	0 reserved
			data continuous bit stream of data for 448 observations, 267 bits per observation with no added bits to end on an even octet
Section 5 (End section)	1-4	14993-14996	7777 encoded CCITT International Alphabet No. 5

Figure 2-3. BUFR message of 448 surface observations using Table
descriptor 3 07 002

D

2.7 Code Tables and Flag Tables. Since some meteorological parameters are qualitative or semi-qualitative, they are best represented with reference to a code table.

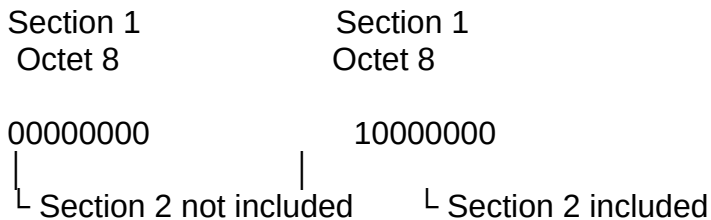
2.7.1 Code Tables. BUFR code tables and flag tables refer to elements defined within BUFR Table B. They are numbered according to the X and Y values of the corresponding Table B reference. For example, the Table B entry 0 01 003, WMO Region number, geographical area, indicates in the Unit column that this is a BUFR code table, the number of that code table being 0 01 003.

Many of the code tables that have been included in the BUFR specification are similar to existing WMO code tables for representing character data. Attachment II of the WMO Manual on Codes, Volume 1, Part B is a list of the code tables associated with BUFR Table B and the existing specifications and code tables of the WMO Manual on Codes, Volume 1, Part A.

There is not a one-to-one BUFR code table relationship to the character code tables. The character Code Table 3333, Quadrant of the Globe, for example, has no meaning in BUFR, as all points on the globe in BUFR are completely expressed as latitude and longitude values.

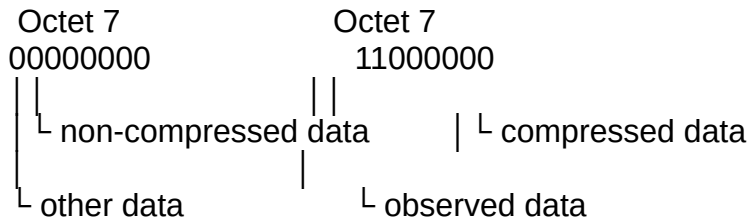
2.7.2 Flag tables. In a flag table, each bit indicates an item of significance. A bit set to 1 indicates an item is included, or is true, while a bit set to 0 indicates omission, or false. In any flag table, when all bits are set it is an indication of a missing value. Flag tables additionally enable combinations to be identified. In all flag tables within the BUFR specification, bits are numbered from 1 to N from most significant to least significant within a data width of N bits, i.e., from left (bit 1) to right (bit N).

2.7.3 Flags. Flags, without reference to a flag table, are also used within Sections 1 and 3 of a BUFR message. In Section 1, octet 8, if bit 1 = 0 this is an indication that the optional section 2 is not contained within the message. If bit 1 = 1, then Section 2 is included.



Similarly, the two flag bits in Section 3, octet 7 have these meanings:

Section 3	Section 3
-----------	-----------



2.8 Local Tables. Since a data processing center may need to represent data conforming to a local requirement, and this data is not defined within Table B, specific areas of Table B and D are reserved for local use (Figure 2-4). These areas are defined as entries 192 to 255 inclusive of all classes. Centers defining classes or categories for local use should restrict their use to the range 48 to 63 inclusive.

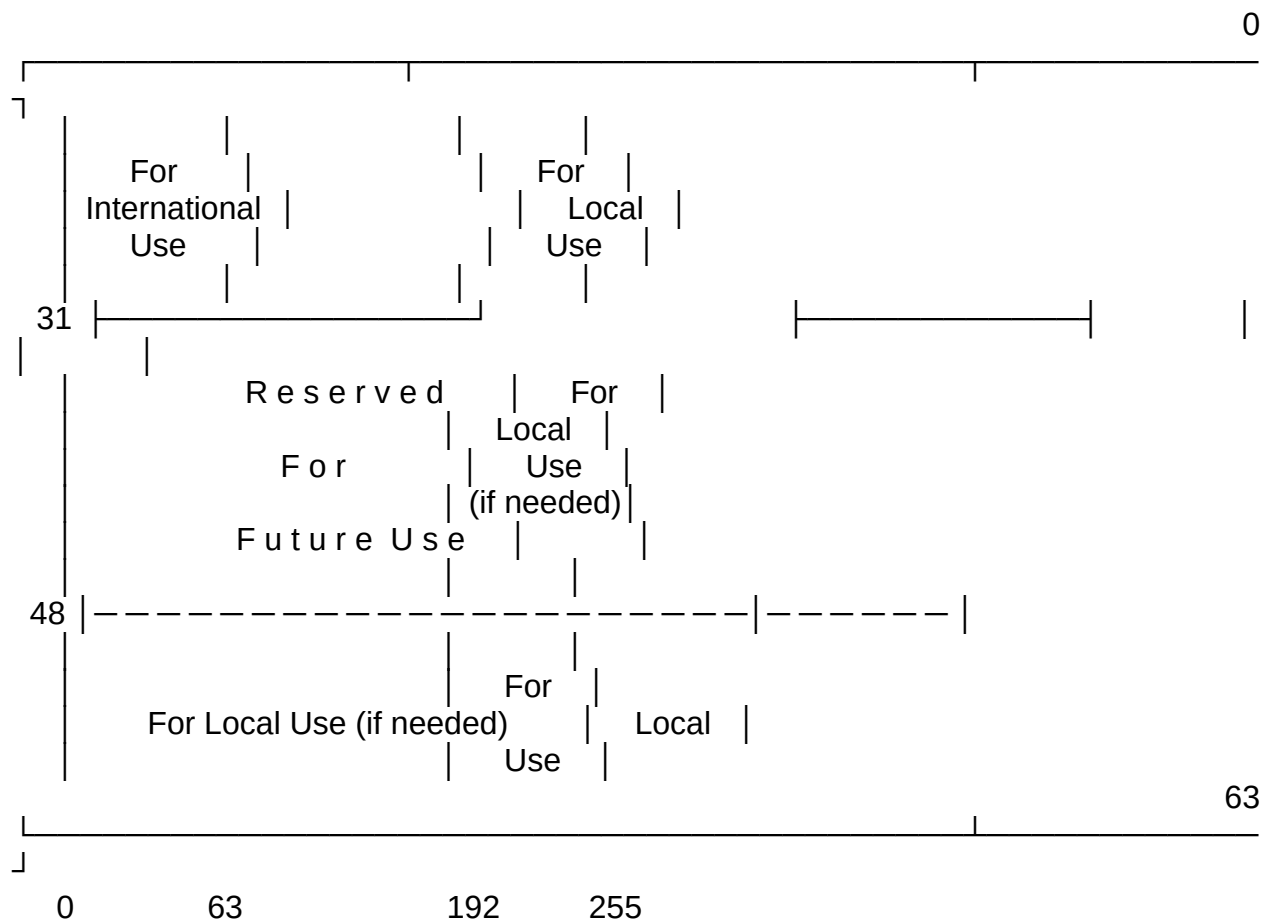


Figure 2-4. Table reservations

If a data processing center had multiple sources of data receipt, for example, it may be necessary to indicate the source of an observation by the circuit from which the data was received. A local Table B descriptor such as

0 54 192 could be used which may be a code table specifying circuits of transmission. The Table B entry could be:

Table Reference	Element Name	Units	Scale	Reference Value	Data Width (Bits)
0 54 192	Circuit	code table	0	0	3

The corresponding local code table could be:

0 54 192

Circuit designators for data receipt

code figure	circuit
0	GTS
1	AWN
2	AUTODIN
3	ANTARCTIC
4-7	Reserved

Using the same Table D descriptor, 3 07 002, as in Figure 2-1, adding the local descriptor 0 54 192 would produce the expansion as in Figure 2-5. The following modifications would have to be made to the BUFR message if the local descriptor 0 54 192 were to be included in a message (Figure 2-6):

Section 0, octets 5-7, the total length of the message, increases from 14996 octets to 14998 octets.

Section 1, octet no. 12 (octet 20 within the message) would have the version number of the local tables in use.

Section 3, octets 1-3, the encoded value would increase from 10 octets to 12 octets. If one descriptor were being added, the length of the section increases by 2 in order to keep the section an even number of octets. Octets 5-6, number of data subsets decreases from 448 to 443. The number of data subsets have been reduced to keep the total message length under the 15000 octet maximum.

Also in Section 3, the descriptors will occupy octets 8-11 vice octets 8-9 to accommodate the added descriptor.

Note that in Section 4, octets 1-3, the encoded value for length of section remains the same at 14956 octets. The number of bits needed for 448 subsets without a local

SECTION 4
WIDTH IN BITS

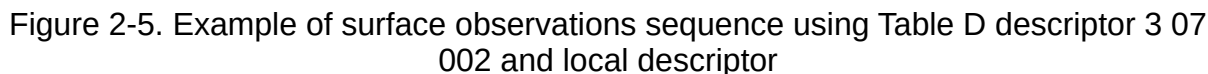


Figure 2-5. Example of surface observations sequence using Table D descriptor 3 07 002 and local descriptor

	Section	Octet in	Encoded		
	Octet No.	Message	Value	Description	
Section 0					
(indicator section)	1-4	1-4	BUFR	encoded international CCITT Alphabet No. 5	
	5-7	5-7	14998	total length of message (octets)	
	8	8	2	BUFR edition number	Section 1
(identification section)	1-3	9-11	18	length of section (octets)	
	4	12	0	BUFR master table	
	5-6	13-14	58	originating center (U.S. Navy	- FNOC)
	7	15	0	update sequence number	
	8	16	0	indicator that Section 2 not included	
	9	17	0	Table A - surface land data	
	10	18	0	BUFR message sub-type	
	11	19	2	version number of master tables	
	12	20	1	version number of local tables	
	13	21	92	year of century	
	14	22	4	month	
	15	23	18	day	
	16	24	0	hour	
	17	25	0	minute	
	18	26	0	reserved for local use by ADP centers (also need to complete	
				even number of octets for Section	3
(Data description section)	1-3	27-29	12	length of section (octets)	
	4	30	0	reserved	
	5-6	31-32	443	number of data subsets	
	7	33	BIT 1=1	flag indicating observed data	
	8-11	34-37	0 54 192	local and Table D descriptors	
			3 07 002	in F X Y format	
	10	38	0	need to complete section with	
				an even number of octets	
Section 4					
(Data section)	1-3	39-41	14956	length of section (octets)	
	4	42	0	reserved	
	5-14956	43-14994		data continuous bit stream of data for 443 observations, 270 bits per observation plus 6 bits to end on even octet	
Section 5					
(End section)	1-4	14995-14998	7777	encoded CCITT international Alphabet No. 5	

Figure 2-6. BUFR message of 443 surface observations using 2 descriptors, local descriptor 0 54 192 and Table B descriptor 3 07 002.

CHAPTER 3

Using Data Replication

3.1 Introduction. When encoding a series of parameters a fixed number of times for all reports represented in Section 4, it may be possible to choose from one of several methods for using Section 3 descriptors.

3.2 Data Replication Examples. If there were 4 elements of cloud information that were described by the Table B descriptors 0 08 002 0 20 011 0 20 012 0 20 013, and these elements were to be repeated 4 times, these 16 total elements of data in Section 4 may be described in the following ways:

1. long and cumbersome method - each element described individually

```
0 08 002 0 20 011 0 20 012 0 20 013
0 08 002 0 20 011 0 20 012 0 20 013
0 08 002 0 20 011 0 20 012 0 20 013
0 08 002 0 20 011 0 20 012 0 20 013
```

2. using the replication operator -

```
1 04 004 0 08 002 0 20 011 0 20 012 0 20 013
```

The meaning of the descriptor 1 04 004 is that the F portion (1) is indicating this is a replication operator, the X portion (04) means the following 4 descriptors are to be repeated Y (004) times.

3. combine replication operator and Table D descriptor

```
1 01 004 3 02 005
```

In this particular example of Table B descriptors there is defined a Table D descriptor 3 02 005 which expands to the 4 descriptors 0 08 002 0 20 011 0 20 012 0 20 013. The replication operator 1 01 004 followed by 3 02 005 means the data in Section 4, defined by the Table D descriptor 3 02 005, is repeated 4 times.

Using either a replication operator followed by a Table B descriptor or a replication operator followed by a Table D descriptor, if it exists, produces the same definition of data as repeating Table B descriptors. Note, in example 3, that the count of the number of descriptors to be replicated (X, 01) applies to the single Table D descriptor that is actually in the message, and NOT to the set of possibly very many descriptors that the single type 3 descriptor represents.

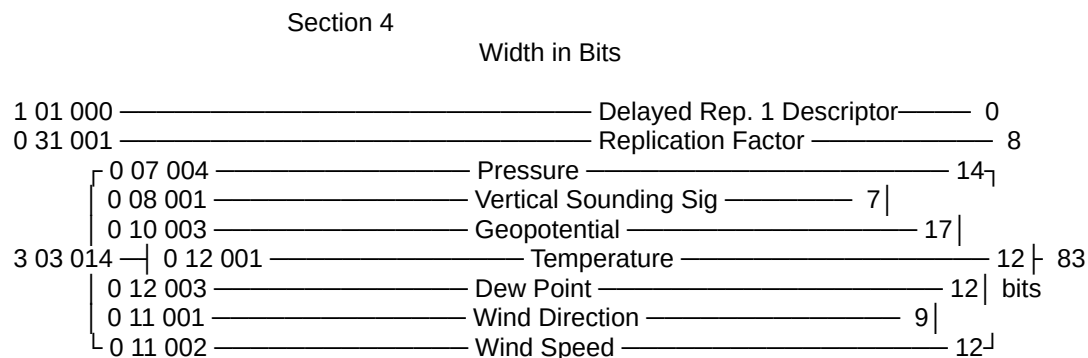
A special form of the replication operator allows the replication factor to be stored with the data in Section 4, rather than with the descriptor in Section 3. This is particularly useful when describing data such as TEMP or BATHY observations where the number of levels differs from observation to observation. The delayed replication operator is of the form F X Y where F = 1, X indicates how many descriptors are to be replicated, and Y = 000. This operator is to be followed by a Table B descriptor from Class 31. The Class 31 descriptor is not included in the count (X) of the number of following descriptors to be replicated. Thus, if the following sequence of descriptors appeared in Section 3: 1 01 000 0 31 001 0 03 014, the meaning of these descriptors is:

1 01 000 F = 1 replication operator
 X = 01 1 descriptor is replicated, not
 counting, i.e. skipping over, the
 0 31 001 descriptor
 Y = 000 delayed replication

0 31 001 F = 0 Table B descriptor
 X = 31 Class 31 - data description operator
 qualifiers
 Y = 001 delayed descriptor replication
 factor
 occupying 8 bits in Section 4 (Table
 B, Class 31 definition)

3 03 014 F = 3 Table D descriptor
 X = 03 Category 03 - meteorological
 sequences common to vertical
 sounding data
 Y = 014 entry 14 of Category 03

The Table D descriptor 3 03 014 expands into seven descriptors. The Section 4 data width for the expansion of 3 03 014 is 83 bits.



For each observation encoded into Section 4 the 8 bits preceding the pressure data indicates how many times the following 7 elements are replicated.

Figure 3-1 is an example of TEMP observations sequence using a single Table D descriptor which expands to include delayed replication. In this example, the replication factor indicates how many levels are contained within the observation. The bit count of 245 bits is for 1 level, each additional level would require 83 bits.

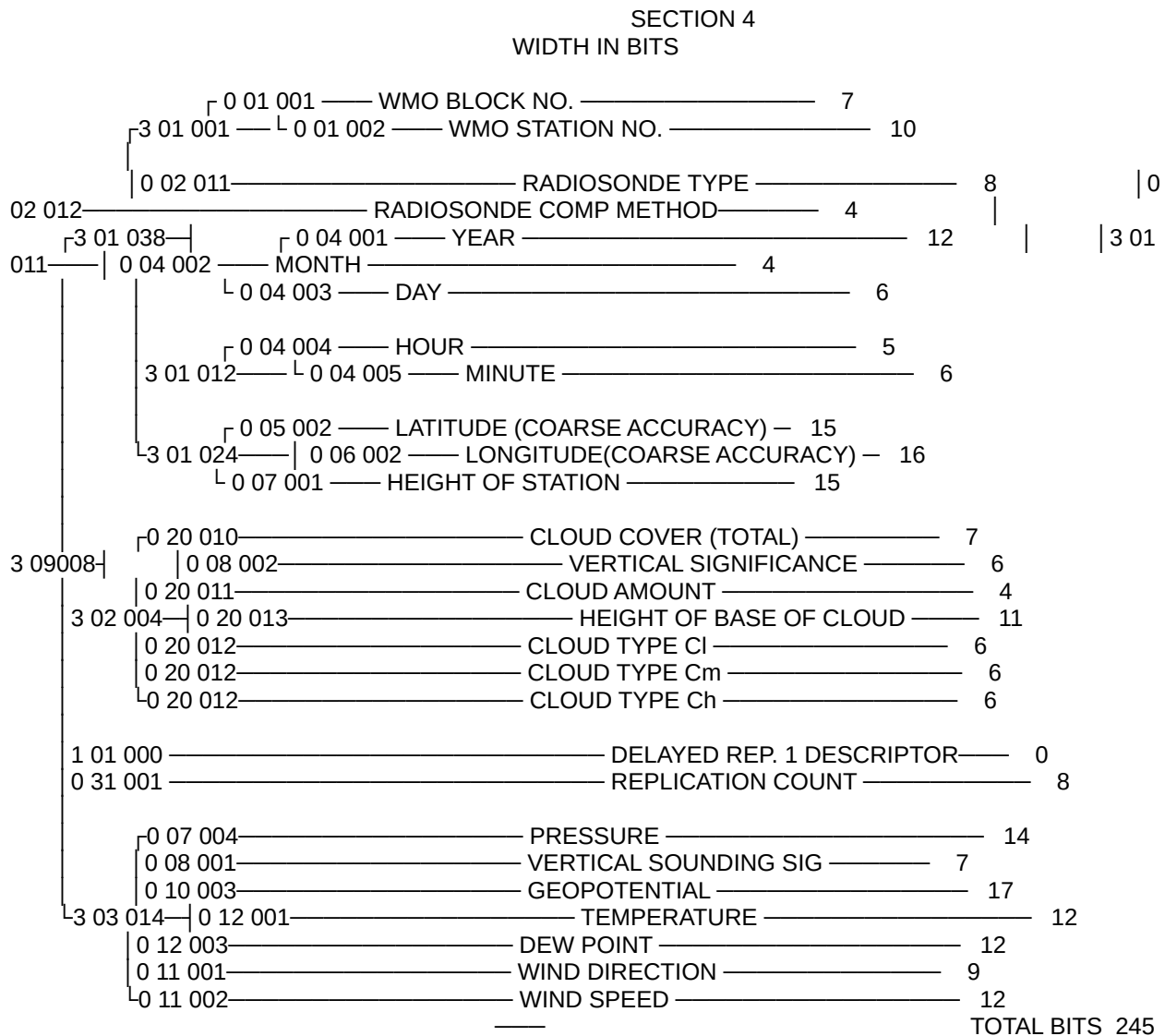


Figure 3-1. Example of TEMP observations sequence using delayed replication

CHAPTER 4

Data Compression

4.1 Introduction. Even though BUFR makes efficient use of space by virtue of binary numbers that take only as many bits as are necessary to hold the largest expected value, a further compression may be possible.

4.2 Method Used for Data Compression. The method employed by BUFR for data compression is similar to that used in the WMO Code FM 92 GRIB (GRidded Binary fields). Like elements from the full set of observations are collected together, their minimum values subtracted out, and the difference from the minimum are then encoded with a bit length selected to hold the largest difference from the minimum value. This is repeated for all the elements.

Using the following group of identically defined data subsets:

	station number	station height	pressure	temperature	dew point
subset 1	101	296	10132	122	110
subset 2	103	291	10122	121	110
subset 3	107	310	10050	105	099
subset 4	112	295	missing	110	102
subset 5	114	350	10055	095	089
subset 6	116	325	10075	101	091

Extraction of the minimum value of each element gives:

101	291	10050	095	089
-----	-----	-------	-----	-----

Each value can now be represented as the difference from these minimum values:

	station number	station height	pressure	temperature	dew point
subset 1	0	5	82	27	21
subset 2	2	0	72	26	21
subset 3	5	19	0	10	10
subset 4	11	4	missing	15	13
subset 5	13	59	5	0	0
subset 6	15	34	25	6	2

After each difference from the minimum value has been determined for each element, determine the number of bits necessary to store the largest of the difference values for each element. For the station number the largest difference is 15 which is equivalent to

1111₂, or 4 bits. However this presents a small problem. All four bits set on, as is the case for the number 15, is properly interpreted as "missing", not as a numeric value of 15. What is done is to simply add one bit to the number needed to store the largest difference value; thus 15 gets stored in 5 bits, as 01111. It is not necessary to add one bit to the bit lengths for all the elements; it is only necessary when one of the numbers to be encoded "fills" the available space; that is, if the number is 3 to be stored in 2 bits, 7 in 3 bits, 15 in 4 bits, 31 in 5 bits, etc. A convenient way to do this and assure that there is always room for "missings" (if needed) is to add 1 to the largest difference value and figure the number of bits based on this larger-by-one value.

In the example, the station height would be placed in 6 bits; the pressure in 7 (with the "missing" indicated as 111111), etc., as in the following table:

	station number	station height	pressure	temperature	dew point
largest difference value +1	16	60	83	28	22
number of bits	5	6	7	5	5

Whereas in the non-compressed storage of data in Section 4 there is a continuous bit stream for all parameters for an entire observation, in the compressed form all elements of the same parameter from each observation form a continuous stream (Figure 4-1). In order to determine what the minimum value is that has to be added back to each of the following elements, and how many bits are being used for the storage of these elements, there are two additional items appearing in the compressed form of storage in Section 4 that do not appear in the non-compressed form.

These items are:

- (1) the minimum value of this parameter and,
- (2) the number of bits that are being used for the storage of each element.

These items of information precede the element values. The Section 4 representation for compressed data for each parameter used in the example above is:

Station number minimum value (101) occupying 10 bits as specified by the Table B data width for entry 0 01 002 followed by:

6 bits containing the count in bits (5) that each of the station numbers will occupy, followed by:
The 6 station number differences from the minimum values (0,

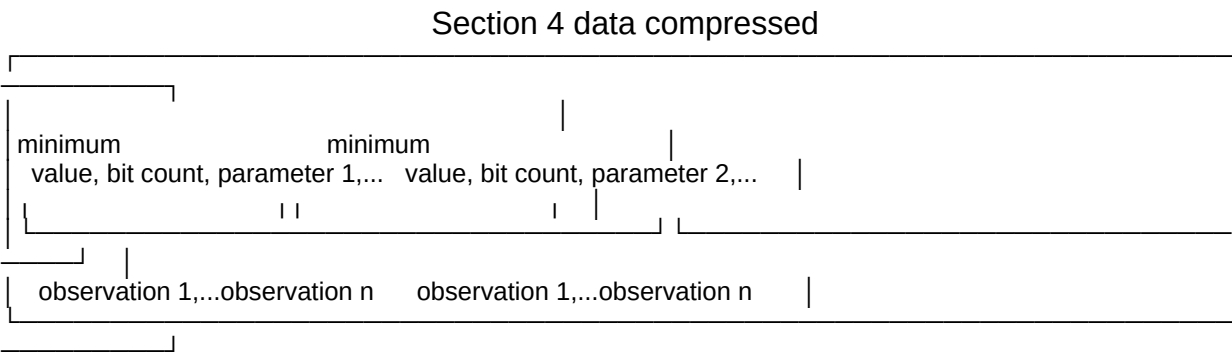
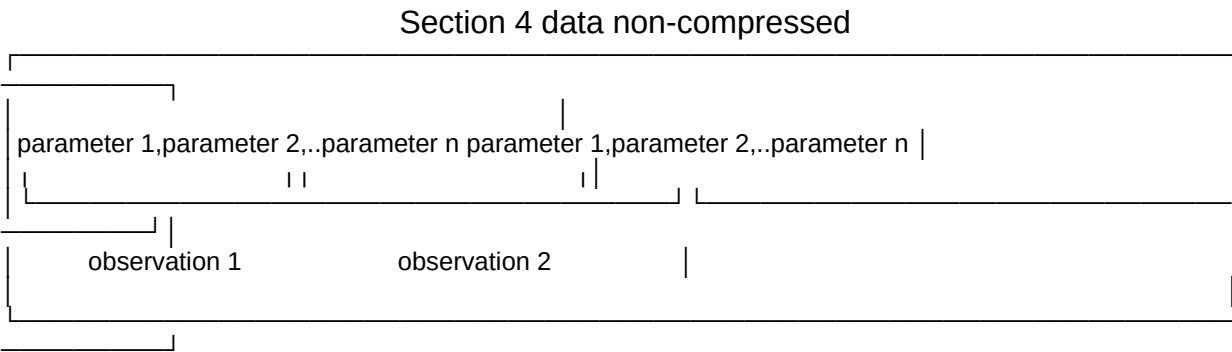


Figure 4-1. Comparison of non-compressed and compressed data in
Section 4

2, 5, 11, 13 and 15), where each value occupies 5 bits.

After the last station number difference (15), the next 15 bits (Table B data width for entry 0 07 001) will be taken by the minimum value for station height (291) followed by the count of bits to represent the differences (6) and then each of the elements occupying 6 bits apiece (5, 0, 19, 4, 59, 34).

Continuing the process for all 5 parameters would produce within Section 4 the following bit counts:

	station number	station height	pressure	temperature	dew point
Table B descriptor	0 01 002	0 07 001	0 10 004	0 12 004	0 12 006
data width to contain minimum value	10	15	14	12	12
6 bits containing bit count of parameter	6	6	6	6	6
Total bits preceding each parameter	16	21	20	18	18
data width to represent difference from minimum	5	6	7	5	5
compressed data representation for 6 subsets	30	36	42	30	30
total bit count for 6 subsets including compression bit counts	46	57	62	48	48 = 261

261 bits are necessary to represent all 6 subsets in compressed form in Section 4.

Using the same set of values for the 6 subsets in non-compressed form there would be bit counts in Section 4 as follows:

	station number	station height	pressure	temperature	dew point
Table B descriptor data width	10	15	14	12	12
total bit count for 6 subsets	60	90	84	72	72 = 378

A total of 378 bits are necessary to represent all 6 subsets in non-compressed form.

There are other conditions that can occur when encoding compressed data. If all elements of a set of parameters are missing, the minimum value occupying the specified Table B data width in Section 4 shall be set to all 1's, the 6 bits specifying how many bits are used for each value will be set to 0, and the difference values will be omitted. If, for example all the dew points were missing from the 6 subsets then the number of bits to represent dew point would be reduced to only include the Table B data width for dew point (12 bits) and the 6 bits specifying the bits used for each value.

	station number	station height	pressure	temperature	dew point					
Table B descriptor	0 01 002	0 07 001	0 10 004	0 12 004	0 12 006					
data width to contain minimum value	10	15	14	12	12					
6 bits containing bit count parameter will occupy	6	6	6	6	6					
Total bits preceding each parameter	16	21	20	18	18					
compressed data (difference from minimum)	5	6	7	5	0					
compressed data representation for 6 subsets	30	36	42	30	0					
total bit count for 6 subsets including compression identifiers	46	+	57	+	62	+	48	+	18	= 231

In the non-compressed form, storage of the missing dew point values would still occupy 12 bits each, with all bits set to 1.

	station number	station height	pressure	temperature	dew point
Table B descriptor data width	10	15	14	12	12
total bit count for 6 subsets	60	+ 90	+ 84	+ 72	+ 72 = 378

The other condition that may occur is if all the difference values are identical, then, the 6 bits specifying the count of bits for each difference value will set to 0, and difference values will be omitted. This condition would produce the same bit count as if all elements were missing.

Set of parameters missing:

minimum value occupying number of bits as indicated in Table B set to all 1's

6 bits specifying how many bits are used for each value set to 0

difference values omitted

Set of identical parameters:

minimum value occupying number of bits as indicated in Table B set to minimum value (actual value for all parameters)

6 bits specifying how many bits are used for each value set to 0

difference values omitted

Data compression is most effective when the range of values for the parameters is small. In the example of the 6 subsets, each parameter has a difference from the minimum value, where the number of bits to represent the difference is half, or less than half, the number of bits required in non-compressed form for storage in Section 4, as indicated by the Table B entry data width. If the 6 subsets were put into a message where compression was not applied, the length of the message would be 100 octets (Figure 4-2). By applying compression, the length of the message would be reduced to 86 octets (Figure 4-3).

Using the range of values for the same 6 subsets, not realistic, but to show the effect of compression for a large data set, a total of 4267 subsets could be put into a BUFR message not exceeding 15000 octets (Figure 4-5). In non-compressed form there would only be 1898 subsets within the 15000 octet limit (Figure 4-4).

	Section	Octet in	Encoded					
	Octet No.	Message	Value	Description				
Section 0								
(indicator section)	1-4	1-4	BUFR	encoded international CCITT Alphabet No. 5				
	5-7	5-7	100	total length of message (octets)				
	8	8	2	BUFR edition number	Section 1			
(identification section)	1-3	9-11	18	length of section (octets)				
	4	12	0	BUFR master table				
	5-6	13-14	58	originator (U.S. Navy - FNOC)	7	15	0	update
sequence number								
	8	16	0	indicator for no Section 2				
	9	17	0	Table A - surface land data				
	10	18	0	BUFR message sub-type				
	11	19	2	version number of master tables				
	12	20	0	version number of local tables	13	21	92	year of
century								
	14	22	4	month				
	15	23	18	day				
	16	24	0	hour				
	17	25	0	minute				
	18	26	0	reserved for local use by ADP	centers			
(also needed to complete				even number octets for section	Section 3			
(Data description section)	1-3	27-29	18	length of section (octets)				
	4	30	0	reserved				
	5-6	31-32	6	number of data subsets				
	7	33	bit 1=1	flag indicating observed data				
			bit 2=0	flag indicating no compression	8-17	34-43	0 01 002	WMO
station no.								
		0 07 001		height of station				
		0 10 004		pressure				
		0 12 004		temperature				
		0 12 006		dew point				
	18	44	0	needed to complete section with	an even			
number of octets								
Section 4								
(Data section)	1-3	45-47	52	length of section (octets)				
	4	48	0	reserved				
	5-52	49-96		data continuous bit stream of data for 6 subsets, 63 bits per subset plus 6 bits to end on even octet				
Section 5								
(End section)	1-4	97-100	7777	encoded CCITT international Alphabet No. 5				

Figure 4-2. BUFR message of 6 subsets in non-compressed form

	Section	Octet in	Encoded	Description				
	Octet No.	Message	Value					
Section 0								
(indicator section)	1-4	1-4	BUFR encoded international CCITT Alphabet No. 5					
	5-7	5-7	86	total length of message (octets)				
	8	8	2	BUFR edition number				
Section 1								
(identification section)	1-3	9-11	18	length of section (octets)				
	4	12	0	BUFR master table				
	5-6	13-14	58	originator (U.S. Navy - FNOC)	7	15	0	update
sequence number								
	8	16	0	indicator for no Section 2				
	9	17	0	Table A - surface land data				
	10	18	0	BUFR message sub-type				
	11	19	2	version number of master tables				
	12	20	0	version number of local tables				
	13	21	92	year of century				
	14	22	4	month				
	15	23	18	day				
	16	24	0	hour				
	17	25	0	minute				
	18	26	0	reserved for local use by ADP				centers (also
needed to complete								
(Data description section)	1-3	27-29	18	length of section (octets)				even number octets for section Section 3
	4	30	0	reserved				
	5-6	31-32	6	number of data subsets				
	7	33	bit 1=1	flag indicating observed data				
			bit 2=1	flag indicating compression	8-17	34-43	0 01 002	WMO
station no.								
			0 07 001	height of station				
			0 10 004	pressure				
			0 12 004	temperature				
			0 12 006	dew point				
	18	44	0	needed to complete section with an even number of octets				
Section 4								
(Data section)	1-3	45-47	38	length of section (octets)				
	4	48	0	reserved				
	5-52	49-82	data	261 continuous bits of compressed data plus 11 bits to end				on even octet
Section 5								
(End section)	1-4	83-86	7777	encoded CCITT international Alphabet No. 5				

Figure 4-3. BUFR message of 6 subsets in compressed form

Section	Octet in	Encoded				
Octet No.	Message	Value	Description			
Section 0						
(indicator section)	1-4	1-4	BUFR encoded international CCITT Alphabet No. 5			
	5-7	5-7	15000 total length of message (octets)			
	8	8	2 BUFR edition number			
Section 1						
(identification section)	1-3	9-11	18 length of section (octets)			
	4	12	0 BUFR master table			
	5-6	13-14	58 originator (U.S. Navy - FNOG)	7	15	0 update
sequence number						
	8	16	0 indicator for no Section 2			
	9	17	0 Table A - surface land data			
	10	18	0 BUFR message sub-type			
	11	19	2 version number of master tables			
	12	20	0 version number of local tables			
	13	21	92 year of century			
	14	22	4 month			
	15	23	18 day			
	16	24	0 hour			
	17	25	0 minute			
	18	26	0 reserved for local use by ADP centers (also needed to complete even number octets for section			
Section 3						
(Data description section)	1-3	27-29	18 length of section (octets)			
	4	30	0 reserved			
	5-6	31-32	1898 number of data subsets			
	7	33	bit 1=1 flag indicating observed data			
			bit 2=0 flag indicating no compression			
	8-17	34-43	0 01 002 WMO station no.			
			0 07 001 height of station			
			0 10 004 pressure			
			0 12 004 temperature			
			0 12 006 dew point			
	18	44	0 needed to complete section with an even number of octets			
Section 4						
(Data section)	1-3	45-47	14952 length of section (octets)			
	4	48	0 reserved			
	5-52	49-14996	data continuous bit stream of data for 1898 subsets, 63 bits per subset plus 10 bits to end on even octet			
Section 5						
(End section)	1-4	14997-15000	7777 encoded CCITT international Alphabet No. 5			

Figure 4-4. BUFR message of 1898 subsets in non-compressed form

Section	Octet in	Encoded	Description			
Octet No.	Message	Value				
Section 0						
(indicator section)	1-4	1-4	BUFR encoded international CCITT Alphabet No. 5			
	5-7	5-7	15000 total length of message (octets)			
	8	8	2 BUFR edition number			
Section 1						
(identification section)	1-3	9-11	18 length of section (octets)			
	4	12	0 BUFR master table			
	5-6	13-14	58 originator (U.S. Navy - FNOG)	7	15	0 update
sequence number						
	8	16	0 indicator for no Section 2			
	9	17	0 Table A - surface land data			
	10	18	0 BUFR message sub-type			
	11	19	2 version number of master tables			
	12	20	0 version number of local tables			
	13	21	92 year of century			
	14	22	4 month			
	15	23	18 day			
	16	24	0 hour			
	17	25	0 minute			
	18	26	0 reserved for local use by ADP centers (also needed to complete even number octets for section			
Section 3						
(Data description section)	1-3	27-29	18 length of section (octets)			
	4	30	0 reserved			
	5-6	31-32	4267 number of data subsets			
	7	33	bit 1=1 flag indicating observed data			
			bit 2=1 flag indicating compression	8-17	34-43	0 01 002 WMO
station no.						
			0 07 001 height of station			
			0 10 004 pressure			
			0 12 004 temperature			
			0 12 006 dew point			
	18	44	0 needed to complete section with an even number of octets			
Section 4						
(Data section)	1-3	45-47	14952 length of section (octets)			
	4	48	0 reserved			
	5-52	49-14996	data 119569 continuous bits of compressed data plus 15 bits to end on even octet			
Section 5						
(End section)	1-4	14997-15000	7777 encoded CCITT international Alphabet No. 5			

Figure 4-5. BUFR message of 4267 subsets in compressed form

CHAPTER 5

Table C Data Description Operators

5.1 Introduction. Table C data description operators (Table 5-1) are used when there is a need to redefine Table B attributes temporarily, such as the need to change the data width, scale or reference value of a Table B entry.

5.2 Changing Data Width, Scale and Reference Value. If data from a DRIFTER observation (FM 18-IX Ext., Report of a drifting-buoy observation) were being encoded into BUFR, there are no Table B entries to correspond to latitude and longitude in thousandths of degrees. The Table B entries for latitude and longitude are high accuracy (hundred thousandths of a degree) and coarse accuracy (hundredths of a degree). There are several possible methods to handle the encoding of latitude and longitude for DRIFTER in thousandths of degrees. One method would be to choose the high accuracy Table B entries for latitude and longitude in hundred thousandths of degrees. There would be no loss of accuracy, but a lot of unused bits for each observation would be encoded in Section 4. The high accuracy latitude requires 25 bits for representation, high accuracy longitude 26 bits. To represent latitude and longitude to thousandths of degrees would require 18 and 19 bits respectively. If the extra bits from using high accuracy were not deemed a concern, this would be the easiest method, but if it were desirable to use only the bits required to represent latitude and longitude in thousandths of degrees, there are two ways for this to be accomplished. First, and the least desirable of any method, would be to create local descriptors for Table B with the appropriate scale and reference values for thousandths of degrees. This is the least desirable method because if the BUFR message were to be transmitted to another center, then the receiving center would have to have available to their BUFR decoder program the correct definition of the local descriptors. The other method would be to use the Table C data description operators 2 01 Y to change the data width of the Table B descriptor for latitude and longitude, 2 02 Y to change the scale and 2 03 Y to change the reference values.

There is now a choice to be made between temporarily changing latitude and longitude from hundredths of degrees to thousandths, or, from changing them from hundred thousandths to thousandths. It doesn't matter which is done, as the only difference between the choices will be the Y operand entries of the data description operators.

If it were decided to change the data width of latitude and longitude from hundredths to thousandths of degrees, what first must be done is to determine how many bits are necessary to represent individually latitude and longitude in thousandths of a degree. The maximum value for latitude to be represented in the

5-1. BUFR Table C - Data Description Operators

Table			
Reference	Operand	Operator Name	Operation Definition

F X

- | | | | |
|------|---|---|---|
| 2 01 | Y | Change data width | Add (Y-128) bits to the data width for each data element in Table B, other than CCITT IA5 (character) data, code or flag tables |
| 2 02 | Y | Change scale | Multiply scale given for each non-code data elements in Table B by $10^{(Y-128)}$ |
| 2 03 | Y | Change reference values | Subsequent element descriptors define new reference values for corresponding Table B entries. Each new reference value is represented by Y bits in the Data Section. Definition of new reference values is concluded by encoding this operator with Y=255. Negative reference values shall be represented by a positive integer with the left-most bit (bit 1) set to 1 |
| 2 04 | Y | Add associated field | Precede each data element with Y bits of information. This operation associates a data field (e.g. quality control information) of Y bits with each data element. |
| 2 05 | Y | Signify character | Y characters (CCITT international Alphabet No. 5) are inserted as a data field of Y x 8 bits in length |
| 2 06 | Y | Signify data width for the immediately following local descriptor | Y bits of data are described by the immediately following descriptor |

data in Section 4 would be based on taking into consideration the also to be changed reference value of -9000. The new reference value will be -90000 to accommodate thousandths of degrees. The maximum value of a reported latitude to be encoded into BUFR bits is 180000. This value is arrived at by a reported latitude of 90.000 North which must then be scaled to 10^3 (also to be changed from 10^2) to retain the desired precision, then subtracting the reference value of -90000, producing 180000. The number of bits to accommodate 180000_{10} is 18. To change the data width of the Table B entry for latitude (coarse accuracy) from 15 bits to 18 bits would require the Table C entry 2 01 131. The Y operand 131 is determined by the Operation Definition of adding Y-128 bits to the data width given for the element 0 05 002. The number 128 is the midpoint between 1 and 255 which is the range of values for the 8 bits of Y. Numbers between 1 and 127 will produce a negative value for changing data width, 129 to 255 a positive value.

The next step would be to change the scale from 10^2 to 10^3 in order to properly decode the reported latitude which will be encoded in Section 4 with 18 bits. The WMO BUFR definition for change scale, "Multiply scale given for each non-code data element in Table B by $10^{(Y-128)}$ ", is referring to the result of 10^{scale} . For Table B entry 0 05 002, the scale is 2. In this case it is the resultant value 100 which is to be multiplied by $10^{(Y-128)}$, not the scale 2. Thus, the data description operator to change the scale for Table B entry 0 05 002 would be 2 02 129.

To complete the necessary changes for Table B, the reference value also needs to be modified from -9000 to -90000. Here again it must be determined how many bits are necessary to accommodate the new value, as the new reference value itself is encoded into Section 4. The number of bits to accommodate 90000 (positive value) is 17. It is, however, necessary to indicate this is to be a negative value which will require an additional bit. To indicate a new reference value as negative, the left most bit of the reference value encoded into Section 4 is set to 1. The sequence of operators needed to redefine or change a reference value is:

- 1) the 2 03 018 "change reference values operator", which announces a change and states how many bits are set aside for the new reference value in the data section (18 in this example)

- 2) one or more regular (F=0) data descriptors to indicate which variable(s) are to have new reference values. There are, of course, as many 18-bit values in the data as there are data descriptors following the 2 03 018 descriptor.

In this particular case it will not be necessary to have separate Data Description operators to modify longitude data width and change of scale. The increase in number of bits for data width to accommodate longitude to thousandths of degrees is also 3. The change of scale also remains the same. There will, however, be a required change of reference value from -18000 to -180000. By following the same steps as when changing the latitude Table reference value, the Data Description operator for changing

the longitude reference value would be 2 03 019 followed by the data descriptor 0 06 002, followed by the descriptor 2 03 255 to indicate the end of the list of descriptors for which reference values are being changed.

Once Data Description operators 2 01 Y, 2 02 Y and 2 03 Y have been used in Section 3, they remain in effect for the rest of whatever follows in the Section 3 data descriptions. To cancel operator 2 01, and 2 02, the additional entries must 2 01 000 and 2 02 000 must be included in Section 3. To cancel the reference value change indicated by the operator 2 03 018, there must be included in Section 3 an operator 2 03 000.

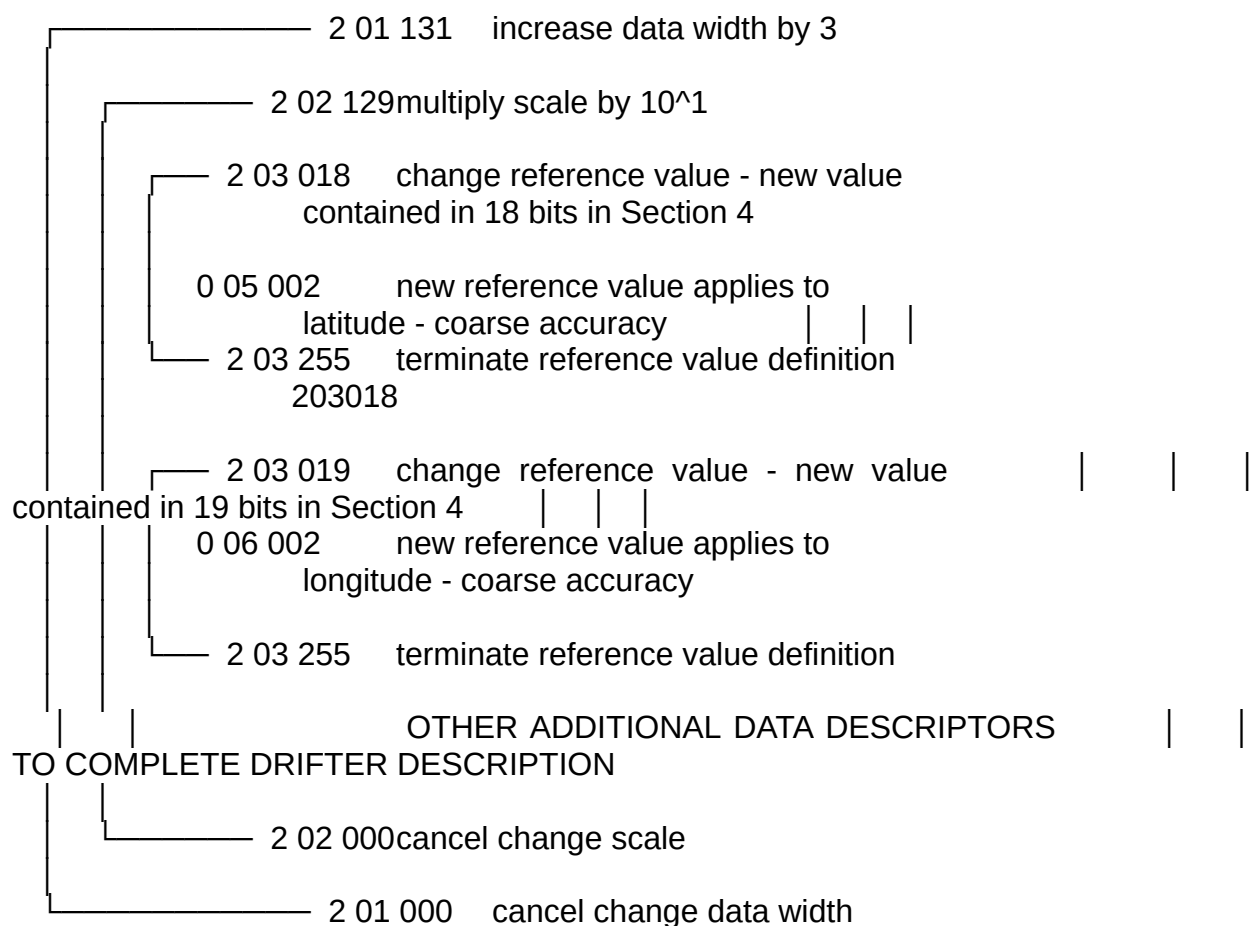
The data description operators encoded into Section 3 for DRIFTER observations would then be:

0 01 005 buoy/platform identifier

0 02 001 type of station

3 01 011 Table D descriptor which expands to
descriptors for year, month and day

3 01 012 Table D descriptor which expands to
descriptors for hour and minute



2 03 000 Cause all redefined reference
values to revert back to standard
Table B values

The order for cancellation of nested Data Description operators follows the above pattern where the last defined is the first canceled.

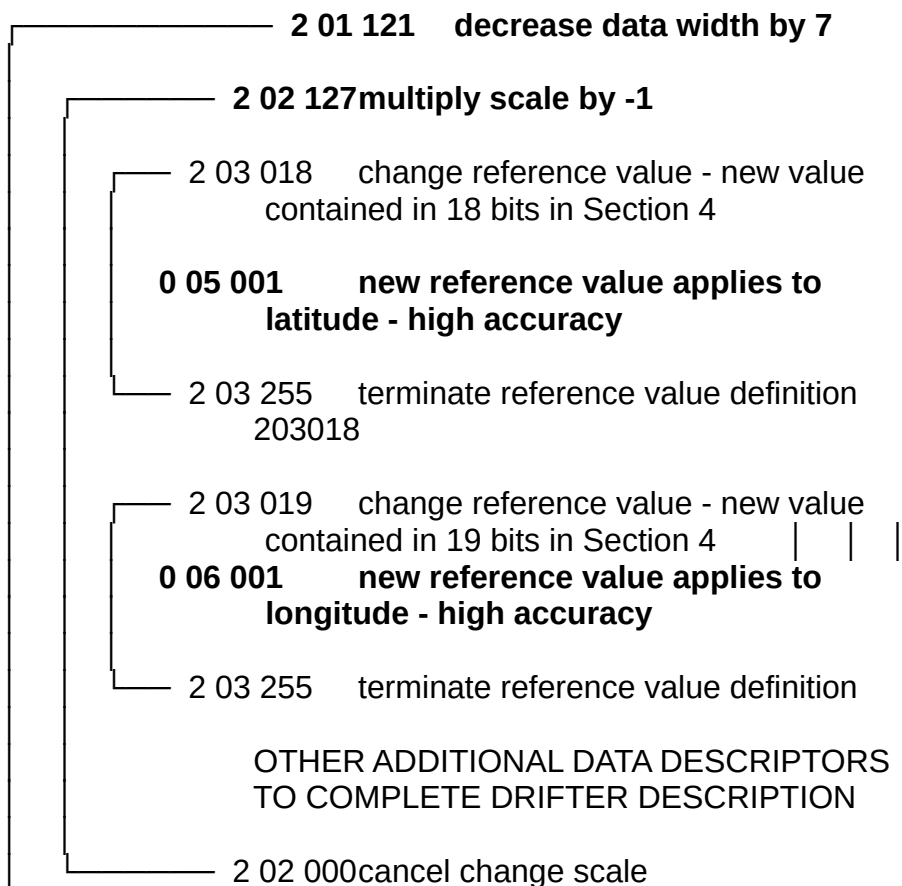
If instead of changing latitude and longitude from hundredths to thousandths, it were to be changed from hundred thousandths to thousandths the following descriptions would be used:

0 01 005 buoy/platform identifier

0 02 001 type of station

3 01 011 Table D descriptor which expands to
descriptors for year, month and day

3 01 012 Table D descriptor which expands to
descriptors for hour and minute



|
|_____ 2 01 000 cancel change data width

2 03 000 Cause all redefined reference
values to revert back to standard
Table B values

Which would be the better of the methods? Again, use of local descriptors to define latitude and longitude is not a good idea as their use may cause a BUFR message to be undecodable in some other center. Of the two other methods, using high accuracy latitude and longitude, or using Data Description operators to change latitude and longitude definitions to thousandths of degrees will each produce the same results. In terms of number of bits saved by changing to thousandths of degrees over high accuracy, a DRIFTER observation containing data equivalent to the DRIFTER code (FM 18-IX Ext. Sections 0 through Section 2) would require 214 bits per observation using high accuracy latitude and longitude. If latitude and longitude were changed by Data Description operators to thousandths of degrees then the observation would require 200 bits per observation, or a savings of 14 bits per observation, hardly worth the effort!

The preceding example does not imply that changing data width, scale and reference values should not be done, but it does point out that to do so to lower the number of bits within the data section for a given parameter is probably not that beneficial. In those instances where the Table B entries do not provide enough significance for new technologies, then the flexibility is provided within BUFR to handle those situations. If, for example, satellites were to measure latitude and longitude to millionths of degrees, then, to maintain significance of those measurements would require changing data width, scale and reference values, at least until (or if) there is a new Table B entry.

This example also shows that when changing data width, scale and reference values, a single Table D descriptor cannot be used in Section 3. The reason is that changing data width and scale apply to all descriptors in Table B until the change data width and/or change scale is canceled. Since the descriptor to be affected may be deep within the Table D expansion process, there is no way to include the Data Descriptor operators in that expansion. A change in reference value, however, can be accomplished while still using a single Table D entry. This is possible because after the entry for change reference value, 2 03 YYY, there must also be included the Table B descriptor or multiple descriptors that are to have new reference values.

5.2.1 Changing Reference Value Only. The Table B entries for geopotential, 0 07 003 and 0 10 003 have a reference value of -400, too restrictive for very low pressure systems. The Table C Data Description operator 2 03 YYY can be placed as the first descriptor in Section 3, followed by the Table B descriptor(s) to which it applies. Placing 2 03 010, followed by 0 10 003 before the Table D descriptor means that each time data is encountered in Section 4 for 0 10 003, the new reference value indicated by the count of 10 bits specified by YYY applies. Within 10 bits the limit of the new reference value as a negative number is -511. The descriptor to conclude the list of descriptors for which new reference values are supplied follows immediately, followed in turn by the Table D descriptor (Figure 5-1). In Figure 5-1, the order of the Section 3 descriptors is:

2 03 010 0 10 003 2 03 255 3 09 008

The Section 4 data will be in the order as indicated by Figure 5-1.

SECTION 4
WIDTH IN BITS

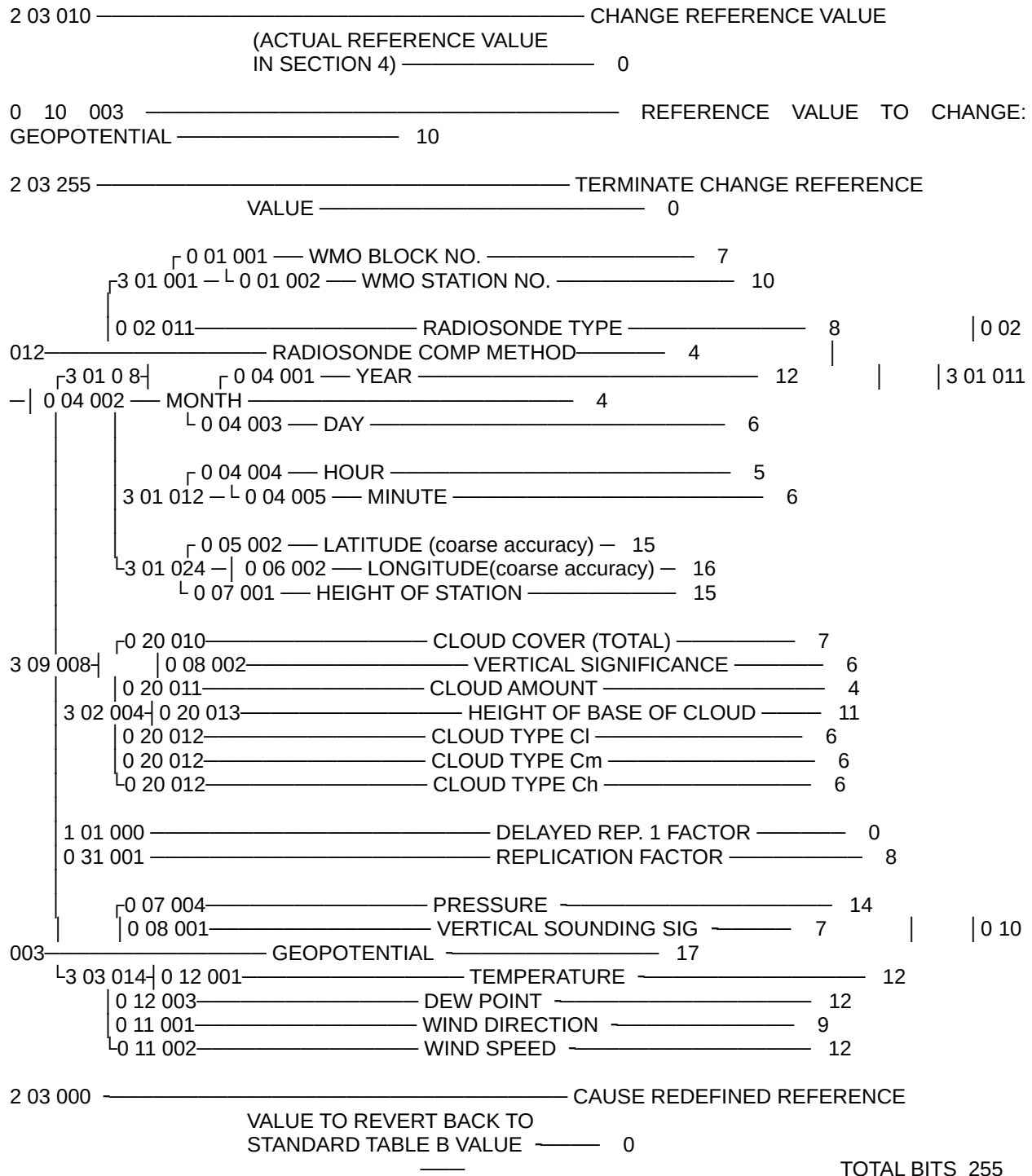


Figure 5-1. Change reference value of geopotential

5.3 Add Associated Field. The Data Description operator 2 04 Y permits the inclusion of quality control information of Y bits attached to each following data element. The additional YYY bits of the associated field appear in the data section as prefixes to the actual data elements. The Add Associated Field operator, whenever used, must be immediately followed by the Class 31 Data Description Operator Qualifier 0 31 021 to indicate the meaning of the associated fields.

0 31 021
Associated field significance

Code
figure

0 Reserved

1 1 bit indicator of quality 0 = good
1 = suspect or bad

2 2 bit indicator of quality 0 = good
1 = slightly suspect
2 = highly suspect
3 = bad

3-6 Reserved

7 Percentage confidence

8-20 Reserved

21 1 bit indicator of correction 0 = original value
1 = substituted/corrected
value

22-62 Reserved for local use

63 Missing value

If quality control information were to be added to a single parameter such as pressure, Table B descriptor 0 07 004, the following sequence would appear in Section 3:

2 04 007 0 31 021 0 07 004 2 04 000

The meaning of this sequence is:

2 04 007 - indicator that 7 bits of data precede all
following Table B entries

0 31 021 - code table entry for the meaning of the 7 bits
preceding the Table B entry

0 07 004 - Table B entry for pressure

2 04 000 - cancellation of the Add Associated Field operator The Section 4 data width for this sequence is 27 bits. The operators 2 04 007 and 2 04 000 do not occupy any bits within Section 4. The 27 bits are taken by 0 31 021 (6 bits) and 0 07 004 (21 bits, 7 bits of associated field plus 14 bits of pressure value)

When multiple Table B entries are preceded by 2 04 YYY as in:

2 04 007 0 31 021 0 07 004 0 31 021 0 10 003 2 04 000

the Add Associated Field operator 2 04 007 and the Data Description Operator Qualifier 0 31 021 both apply to the Table B descriptors 0 07 004 and 0 10 003. The Section 4 data width for the sequence is then:

2 04 007	0 bits
0 31 021	6
0 07 004	21 (7 associated bits plus bits 14 data)
0 31 021	6 (change meaning of associated field)
0 10 003	24 (7 associated bits plus 17 bits data)
2 04 000	0

Note that the associated fields are not prefixed onto the data described by 0 31 YYY descriptor. This is a general rule: none of the Table C operators are applied to any the Table B, Class 31 descriptors.

If quality control information were to be added to the following sequence of parameters as described by the Table D descriptor

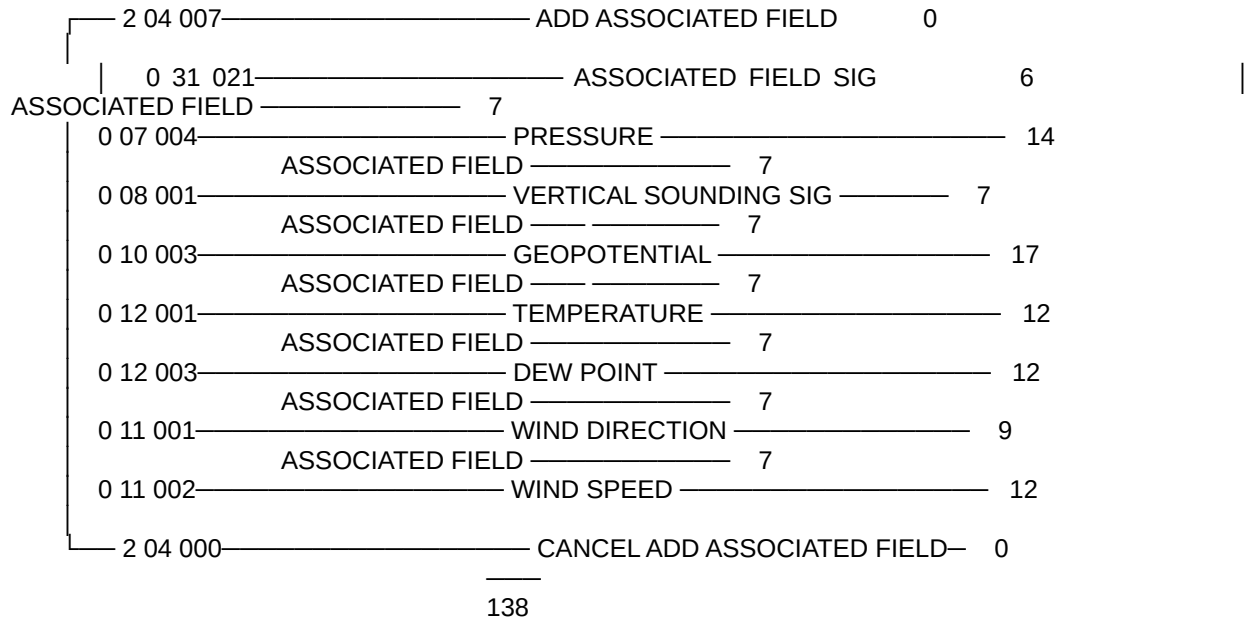
3 03 014:

SECTION 4 WIDTH IN BITS		
3 03 014	0 07 004	PRESSURE 14
	0 08 001	VERTICAL SOUNDING SIG 7
	0 10 003	GEOPOTENTIAL 17
	0 12 001	TEMPERATURE 12
	0 12 003	DEW POINT 12
	0 11 001	WIND DIRECTION 9
	0 11 002	WIND SPEED 12
83		

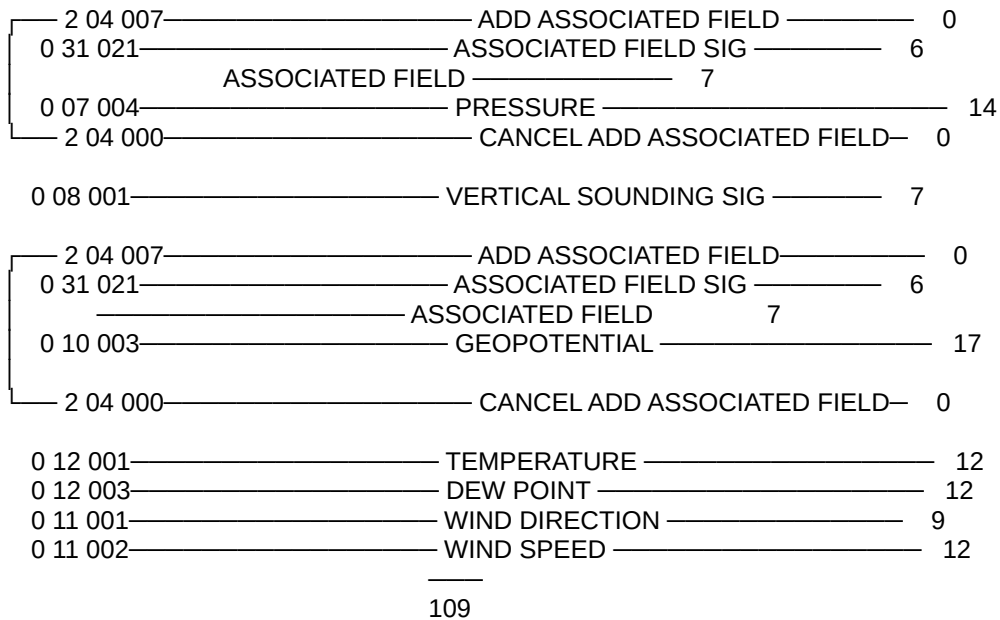
By placing in Section 3 the operators 2 04 YYY and 0 31 021 immediately preceding 3 03 014, and the cancellation operator 2 04 000 following 3 03 014, the following sequence would be produced:

SECTION 4

WIDTH IN BITS



Adding associated fields to a data sequence that is described by a Table D descriptor means the associated fields are placed before all data items in the sequence. If quality control information were to be applied only to the pressure and geopotential parameters, the Table D descriptor could not be used but instead each individual parameter would have to be listed in Section 3.



If quality control information were to be add to TEMP observations as described in Figure 3-1 the following adjustments would have to be made. The single Table D descriptor 3 09 008 could no longer be used as the expansion includes the additional

Table D descriptor 3 03 014 which further expands to those parameters where quality control information would need to be inserted. The actual order of the Section 3 descriptors would now be (Figure 5-2):

3 01 038	3 02 004	1 13 000	0 31 001	2 04 007	0 31 021
0 07 004	2 04 000	0 08 001	2 04 007	0 31 021	0 10 003
2 04 000	0 12 001	0 12 003	0 11 001	0 11 002	

SECTION 4
WIDTH IN BITS

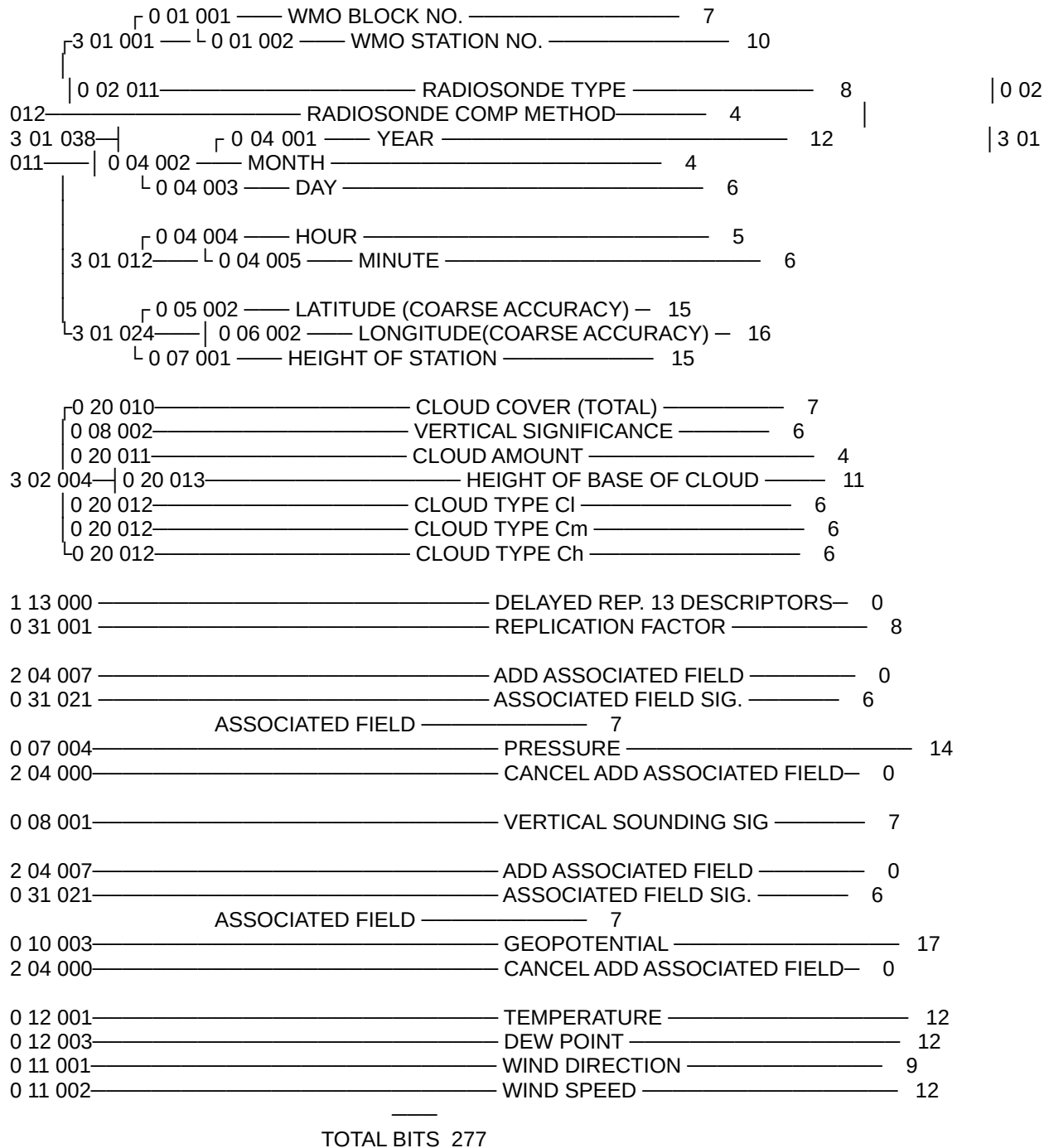


Figure 5-2. Example of TEMP observations sequence using delayed replication and quality control information

5.4 Encoding Character Data. There may be occasions when it is necessary to encode character data into BUFR. An observation encoded into BUFR that originated from the character code FM 13-IX Ext. SHIP, for example, has within that code form the optional inclusion of plain language. If this character information were carried over for encoding into BUFR, the Data Description operator 2 05 Y would be used in Section 3 to indicate the inclusion of character data in Section 4 of the BUFR message. The Y operand of the Data Descriptor indicates the number of characters, encoded CCITT International Alphabet No. 5, inserted as a data field in Section 4.

The following parameters from the FM 13-IX Ext. SHIP code form:

$$\left(\begin{array}{c} 6I_sE_sE_sR_s \\ \text{or ICING} + \\ \text{plain language} \end{array} \right)$$

described by BUFR descriptors would be:

0 20 033 cause of ice accretion
 0 20 031 ice deposit (thickness)
 0 20 032 rate of ice accretion

It would have to be determined in advance how many characters would be allowed for the plain language. If only the word ICING were to be placed in Section 4, the Data Descriptor 2 05 005 would be used. If it were determined that ICING plus 25 additional characters, including spaces, were to be described then the descriptor would be 2 05 030. The data descriptors and data width in Section 4 would then be:

	data width in bits	
0 20 033 cause of ice accretion	4	0 20 031 ice deposit
(thickness) 7		
0 20 032 rate of ice accretion	3	
2 05 030 character information	240	

Since an observation in FM 13-IX EXT. SHIP code would have either the parameters for ice reported, or ICING + plain language, but not both, then if there were no plain language the character information would be set to spaces. If the ICING + plain language were reported then the data for descriptors 0 20 033, 0 20 031 and 0 20 032 would be set to missing, all bits set. Since Section 3 indicates a count of how many subsets (observations) are included in Section 4, the above descriptors apply to all subsets, even if an individual observation does not contain any icing information. In that

case the entire set of icing data for an observation would be set to missing and spaces.

5.5 Signifying Length of Local Descriptors. Local Descriptors were provided in BUFR to enable a data processing center the capability of describing information of any type within BUFR for the center's internal use (Figure 2-4). There does exist, however, the possibility that once data is described in BUFR it may be necessary to transmit a BUFR message to another center, where the BUFR message would contain local information. Since a receiver of the BUFR message may or not know the meaning of the local descriptor, it could be impossible to be able to decode the message, as the receiver would not know the data width in Section 4 of the local information (Figure 2-5). While it could be argued that BUFR messages containing local information should never be transmitted to another center, it may require a separate set of software to remove local information before the message is ready for transmission. To overcome this situation the Data Description operator 2 06 Y was developed to allow local information to be contained within a transmitted message and to give information to the receiver that indicates the length in bits of the local data. The meaning of the Data Description operator 2 06 Y is that the following local descriptor is describing Y bits of data in Section 4 (Figure 5-3). Knowing the width in bits of data in Section 4 then allows the receiver of the message to bypass that number of bits and allow proper decoding of Section 4.

The operator 2 06 Y can only be used when it precedes a local descriptor with $F = 0$. While it is within the rules of BUFR to create local descriptors with $F = 3$ (sequence descriptor), the Data Description operator 2 06 Y cannot be used to bypass whatever number of bits are being described by a sequence descriptor. Since a sequence descriptor expands to other descriptors and in the expansion process other local descriptors or delayed replication may be encountered, there is no way of knowing in advance how many total bits are covered by a sequence descriptor.

SECTION 4
WIDTH IN BITS

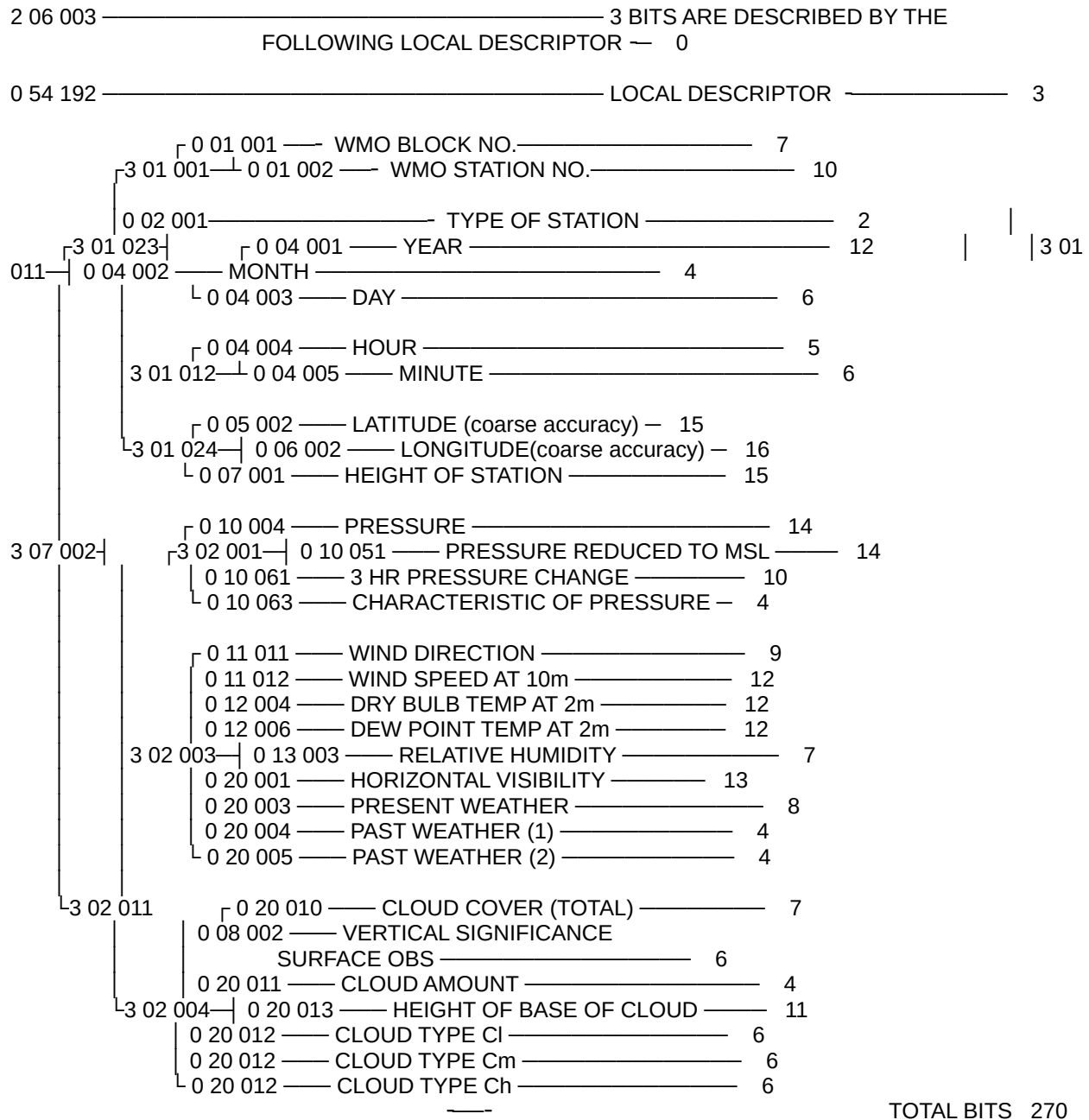


Figure 5-3. Example of surface observations with local descriptor
and data descriptor operator 2 06 Y

Chapter 6

Quirks, Advanced Features, and Special Uses of BUFR

J.D. Stackpole

6.1 Introduction. This chapter is a slightly disparate collection of odds and ends about BUFR: it discusses some of the advanced features that are sometimes overlooked in a casual reading of the WMO Manual, some of the special uses to which data represented in BUFR has been (or can be) put, and offers a fuller explanation of some of the rather obscure portions of the WMO description of the data representation system.

It also details some of the conventions adopted on an ad hoc basis in those (few) cases where the current specifications of BUFR are a little bit ambiguous. It is expected that what is described in this context will find its way into the published specifications all in good time.

In part, this chapter is necessary because it is turning out, with experience, that BUFR is indeed a very powerful data representation system. As people work with the system, they recognize new possibilities that were not thought of in the original design. Sometimes these new possibilities fit right in to the existing system, as though they were implicitly present from the beginning, othertimes they require a slight (or not so slight) augmentation of the BUFR rules and/or descriptors to implement the ideas. The latter must be done with care, of course, so as not to build any (violent) inconsistencies into BUFR. Some of the more promising proposals for change are discussed in this chapter, but are clearly indicated as such.

Also, this chapter is (unfortunately) necessary because some of the features (advanced or not) of BUFR are none too clearly spelled out in the necessarily limited confines of the WMO Manual. Experience has shown that some of the rules and regulations get overlooked and/or misinterpreted in their application. It is hoped that this chapter, and this Guide in general, will help to alleviate these sorts of problems.

BUFR sets out to do a lot; this, in turn, does lead to complexity. There is no free lunch.

As an organizing structure, each Section of a BUFR message/record will be dealt with in their regular order.

6.2 Section 0 - Indicator Section.

6.2.1 Edition Number Changes. There hasn't been any particular difficulty with this section except perhaps for the "Edition Number", currently 2, of the BUFR system. The Edition Number will change only if there is a structural change to the data representation system such that an existing and functioning BUFR decoder would fail to work properly if given a "new" record to decode. A change or augmentation to Tables A, B, D, or the code and flag tables would not involve defining a new Edition for BUFR; one would, of course, be required to change corresponding tables in a computer program but the logic of the program would not have to be changed. Changing tables is easy; changing program logic is not so easy. The former is, indeed, what BUFR is all about.

Edition changes can come about in three main ways. For one, if the basic bit or octet structure of the BUFR record was changed, by the addition of something new in one of the "fixed format" portions of the record, say, this would obviously require computer program changes to work properly. The change from Edition 1 to 2 involved just such a change - see the remarks in Section 1.2.1. These changes are expected to be kept to a bare minimum by the WMO community.

A second way that an edition change can come about is if the data description operators, in Table C, are augmented. These operator descriptors are qualitatively different from simple data descriptors: where the data descriptors just passively describe the data in the record, the operator descriptors are, in effect, instructions to the decoding program to undertake some particular action - just what actions are possible are those defined by Table C. Descriptors of type 1 (F=1), the replication operators, are also in this category - they tell the computer program to do something - but there is little room for change as they are currently defined. Clearly, if some new (and presumably useful) "operation" is defined, by inclusion of an operator in Table C, any decoding programs will have to be modified to respond properly. The descriptor 2 06 YYY (the "skip local descriptor" operator) was one such addition made in the conversion from Edition 1 to Edition 2.

Unfortunately, not all of the "operator" descriptors are collected in Table C. Some of the nominal data descriptors, in particular the "increment" descriptors found in Table A, Classes 4, 5, 6, and 7, take on the character of operators in conjunction with data replication (Regulation 94.5.4) and the operator qualifiers in Table A, Class 31. This will be expanded on further below. However, it is clear that changes or augmentations to the general process of replication, including increments, would involve defining a new Edition of BUFR.

A third change that would require a new Edition would be a change of the Regulations and/or many of the various notes scattered through the documentation. (The "notes", by the way, are as important as the "Regulations" in formally defining BUFR - they contain many of the details that flesh out the rather sparse regulations. Ignore them at your peril.) This is not particularly likely to happen - more likely will be clarifications to the Regulations or notes that will serve to make the rules more precise in (currently) possibly ambiguous cases. This may result in a tightening of a rule (or an interpretation) that may require a current "inappropriate" practice to be eliminated; whether this should be considered as requiring an Edition number change is a matter of some judgment. The WMO will be the final arbiter.

6.2.2 Maximum Size of BUFR Records. As noted elsewhere, there is no theoretical limit to the size of a BUFR message. The largest that can be accommodated by Octets 5-7 would be almost 17 mega-octets (megabytes) but a single bulletin of that size would be a bit much for the WMO Global Telecommunications System (GTS). By general

international agreement single messages should be kept to less than 15,000 octets (15 kilobytes); 10,000 octets is a good safe number to use to be assured that GTS switching centers won't inadvertently truncate the bulletins as they pass them on. A still "experimental" (in WMO terms) feature called BLOK will soon be available to break up large BUFR (and GRIB) records into sizes that the GTS can handle without difficulty. It is better, however, that such large records not be generated in the first place.

6.3 Section 1 - Identification Section.

6.3.1 Master Tables, Version Numbers, and Local Tables.

At present there are no (known) Master Tables for BUFR other than the meteorological set published in the WMO Manual On Codes. That is not to say that such could not exist. That is one of the major strengths of BUFR: any scientific discipline interested in transmitting, storing, or even data basing information unique to it can define its own set of Tables and take advantage of meteorological experience in using the BUFR system.

As is noted elsewhere in this document, only the upper left portion of the (Class by Entry) matrix of descriptors has been defined in the current Master Table B - Classes 00 through 31, Variable number of entries in each class - in the current WMO documentation. Classes 48 through 63 are for local use - this means that any group may define anything they please for those classes; the same is true for Entries 192 through 255 in any Class. The other classes, and whatever unused entries are not spoken for in each class, are set aside for future international usage. Some of the Classes, Class 2 - Instrumentation in particular, are getting alarmingly crowded.

Elements can be added to the international portion of the tables on rather short notice by eliciting the coordinating cooperation of the WMO Working Group on Data Management (WGDM), Sub-Group on Data Representation and Codes (SGDRC). International notification of such additions is accomplished by the World Weather Watch (WWW) Monthly Operations Letter. The WMO body that is parent to the WGDM, the Commission on Basic Systems (CBS), meets every two years or so and, upon CBS approval, the additions to the tables will be published by the WMO. At that point the Tables acquire a new version number. At present the Tables stand at Version 2. This relatively informal method of adding to the tables is possible because the BUFR community is, at present, rather small. It is also possible because of the agreed upon convention that ONLY additions will be made to Tables B or D by this method, descriptors will neither be deleted nor changed, thus existing messages and decoding tables will not be effected as long as they have no need to make use of the new data descriptors.

The SGDRC meets from time to time to study and recommend changes that may involve the structure of BUFR or more substantial changes to the Tables, such as the addition of new operator descriptors or the possible elimination of old and

unused descriptors. This latter step will be taken with great care, however, so as to not make old archives of BUFR data inaccessible. Such recommendations will wend their way through the WMO system, eventually appearing as new Versions of the Tables, upon approval of the CBS. Because the Version number of the Tables is part of the BUFR message, it is only a bookkeeping device for a decoding program to note the Version number and then extract the appropriate Table version from some computer files. The WMO publications will always contain the latest Version of the Tables; it is up to the various meteorological computer centers to maintain their own files of previous versions as well as their own local tables, of course.

The Local portions of the Tables can be updated, changed, augmented, etc. at will by the local group concerned. No international notice is required or expected. It is presumed that bulletins containing local descriptors will not be sent out internationally (but see the discussion of descriptor 2 06 YYY for an exception).

"Local", although not defined in the BUFR documentation, is generally taken to mean "within the processing center that is generating the BUFR messages", and not necessarily one country. The U. S. has a number of processing centers (the civilian weather service, Air Force, Navy, and other groups as well, each potentially identified by a unique processing center number) each one of which is free to use the "local" portions of the BUFR tables as they see fit.

6.3.2 Originating Center (or Centre). The rather arcane method of calculating the number of the originating center described in the Manual arises out of a little history. GRIB (FM 92) was developed first and adopted a pre-existing WMO table of meteorological centers. It is a list of mainly large world and regional meteorological centers that could be expected to have the computer facilities required to generate GRIB bulletins if they had occasion to do so. When BUFR was developed it was realized that observational data could originate from far more locations than the GRIB table could accommodate. Since, in turn, it was recognized that the vast majority of such meteorological data originating locations were already identified by the International Civil Aviation Organization (ICAO), it made the task of identifying BUFR originating centers easy. The algorithm was then developed to convert the ICAO identifier into a unique number that fit within the two octets of space available.

ICAO Document 7910, containing the four-letter code ICAO "Location Indicators" is available from:

Document Sales Unit
International Civil Aviation Organization
1000 Sherbrooke St. West, Suite 400
Montreal, Quebec

Canada H3A 2R2

The price was (US) \$16.50 in 1990 - it is probably more now. The same information (or a subset of it) can be found in WMO Publication 9, Volume C. The ICAO location identifier also forms the "CCCC" part of the WMO standard Abbreviated Heading for all weather messages, as described in Publication 386. Note the rule that if there is a GRIB Table 0 entry already in place for an originating center, that same number should be used for BUFR data messages generated at and sent from that location.

6.3.3 Update Sequence Number. This feature does not seem to have wide use, as yet, but it is a powerful one. Note that the rule does require one to re-send an entire message if even only one element in the message is a correction of a previous message element. The "associated field" (see more on this later) is used to indicate which element(s) is(are) the corrected one(s) within the total message.

6.3.4 Optional Section 2. This section is not usually sent in international messages but it is put to use in some computer centers that use BUFR, frequently in a data base context. Some samples are given below. If it is present, the flag in octet 8 must be set, of course.

6.3.5 BUFR Message Sub-Type. This is purely a local option. As an example here are the sub-types currently in use at the National Meteorological Center, Washington. This sort of information is useful in processing the observational data after it has been decoded from BUFR. By knowing ahead of time, so to speak, in considerable detail just what sort of data is in a BUFR message, it can make the choice of subsequent processors that much easier. It also makes it possible to search through a collection of various data types, encoded in BUFR, and select out only those for which there is a special interest. This has obvious applications in a data base context.

BUFR Data Category 0: Surface data - land

Data Sub-type	Description
0	Unassigned
1	Synoptic - manual
2	Synoptic - automatic
3	Aviation - manual

- 4 Aviation - AMOS
- 5 Aviation - RAMOS
- 6 Aviation - AUTOB
- 7 Aviation - ASOS
- 8 Aviation - METAR
- 9 Aviation - AWOS

BUFR Data Category 1: Surface data - sea

Data Sub-type	Description
0	Unassigned
1	Ship - manual
2	Ship - automatic
3	Drifting buoy
4	Moored buoy
5	Land based C-MAN station
6	Oil rig or platform
7	Sea level pressure bogus
8	Moisture bogus
9	SSMI

BUFR Data Category 2: Vertical soundings (other than satellite)

Data Sub-type	Description
0	Unassigned
1	Rawinsonde - fixed land
2	Rawinsonde - mobile land
3	Rawinsonde - fixed ship
4	Rawinsonde - mobile ship
5	Dropwinsonde
6	Pibal
7	Profiler

BUFR Data Category 3: Vertical soundings (satellite)

Data Sub-type	Description
0	Unassigned
1	Geostationary
2	Polar orbiting
3	Sun synchronous

BUFR Data Category 4: Single level upper-air (other than satellite):

Data Sub-type	Description
0	Unassigned
1	Aircraft - manual
2	Aircraft - reconnaissance
3	Aircraft - automatic (ASDAR)
4	Aircraft - automatic (ACARS)
5	Aircraft - automatic (AMDAR)

BUFR Data Category 5: Single level upper-air (satellite):

Data Sub-type	Description
0	Unassigned
1	Cloud-tracked winds
2	Water-vapor-tracked winds

6.3.6 Date/Time. The Manual suggests placing the date/time "most typical for the BUFR message contents", whatever that may mean, in the appropriate octets. Obviously for synoptic observations the nominal synoptic time is appropriate. But note that the exact time of the observation can be placed in the body of the message if this is of interest or value to

the users of the data. Not only that, but a collection of observation times (and exact locations) could be incorporated into one observation to indicate, for example, the times (and places) that a radiosonde balloon reached particular levels in the atmosphere. This possibility is getting serious attention as very fine mesh numerical models with frequent analysis update cycles are coming into operations. A RAOB can take an hour or more to complete its flight, and travel 40 or 50 km (or more) downwind in that time. That is clearly enough to place the high level parts of the observation into both the next analysis update cycle and at a neighboring gridpoint. Reporting this level of detail would require a major revision to the character based TEMP Code (FM 35) but BUFR can accommodate this additional information with no change whatsoever. [End of commercial for BUFR!]

Collections of satellite observations, which are inherently asynoptic, by convention will have the time of the first observation of the collection in the date/time octets. The exact times for each observation will, of course, be in the body of the message.

6.3.7 "Reserved for use ...". Here again is a playground for the local center. It is not expected that international BUFR messages will contain anything past octet 18 (and that octet will be all zeros per the rule that all Sections have an even number of octets) but there is no real damage if Section 1 is "extended" past octet 18. That is because the "Length of Section" in octets 1-3 will (should) indicate the full size of the section. Any operational decoding program worthy of the name will check the number in octets 1-3 and respond accordingly, presumably by skipping the extra material.

6.4 Section 2 - Optional Section - Examples of Data Base Keys.

6.4.1 U. S. National Meteorological Center Usage. At the U.S. National Meteorological Center (NMC) the Optional Section is being used, internally, as a very simple data base key. The actual data are stored in data subsets (see below), i.e., individual observations. For each observation/subset there is a short collection of information in Section 2, which looks like this:

Content	Element Size
Displacement from start of BUFR message to start of subset (in units of octets)	2 octets
Latitude	2 octets
Longitude	2 octets
Day & hour	2 octets

Identification

6 octets

The first of these 14 octet packets starts in octet 5 of Section 2, with the others following without any break. This rather minimal set of information is enough to select out individual observations using location and/or time criteria. It is not necessary to decode any of the observations to find the desired ones - the displacement count tells you where to go to get each observation.

The alert reader will have noted a difficulty with the above scheme: in the BUFR system there is no requirement that data subsets each start on an exact octet or word boundary; indeed it is rather unlikely that they would, given the essentially random nature of the bit lengths used to store data elements. Yet the "displacement" is specified in terms of octets. Some sort of padding is clearly necessary, so that as the BUFR record is constructed each subset will start on a word (or half-word, or octet) boundary in whatever machine is in use. The actual padding is easy: one simply invents a local descriptor (NMC uses 0 63 255) which is specified to describe 1 bit of padding in the data section without assigning any other "meaning" to the bit. Then one places a delayed replication descriptor (1 01 000, with its associated 0 31 001 count descriptor) in front of the pad descriptor, with the delayed count giving the number of bits inserted to generate a pad of the proper length. This works but leaves one with local descriptors imbedded in the message - a problem if the message is to be sent out non-locally at some future time. It could be expensive to go through the record, remove the padding, and reconstruct a "pure" BUFR record for all the data.

But this can be resolved with the use of the "skip local descriptor" descriptor, 2 06 YYY. Just place it before the local "pad" descriptor, change the XX of the delayed replication descriptor to a value of 2, and the padded record can then be sent out without causing any problems for recipients. The whole thing would look like this:

Descriptors Values

Here is a fragment from
an uncompressed BUFR
record (ignore blank lines)
ddd1 vvv1
ddd2 vvv2
ddd3 vvv3
end of "real" data subset -----> ddd4 vvv4

Delayed rep. of two 1 02 000 -
descriptors n times; 0 31 001 n
n is the number of bits in

the pad, which follows the
8 bits containing the n value

Skip local descriptor	2 06 001	-
Local pad descriptor	0 63 255	(one bit)

And that does it.

Another solution, of course, to the padding problem to create a new international padding descriptor. But since "padding" is machine dependent it seems better to leave the padding up to the local center and not make a regular practice of exchanging padded BUFR messages.

6.4.1.1 BUFR as a Data Base Storage System.

Once the observations/subsets are lined up on octet (or word) boundaries it becomes quite feasible to use BUFR records as a (simple) data base storage format. One restriction applies: all the data subsets must be the same size (i.e., no delayed replications - see below) and not be compressed. A common use of a data base system is to extract one particular data element, temperature, say, from all the available observations, for specific time and geographic ranges. To do so with "lined up" BUFR records all that is necessary is to decode the first subset and take note of the relative location of the temperature data in that subset. Then one simply extracts the temperature information from the relative location in the other subsets without having to (expensively) unpack the entire records.

Of course, this does not allow for all the features of a full relational data base management system. But it may well be sufficient for some more limited uses. It does have the advantage that data can be shared from center to center, and used in similar data base systems, without the necessity of decoding the data (or extracting it from an RDBMS) and re-encoding the data to transmit it in a reasonably efficient format. It already is in a reasonably efficient transmission format. It may be necessary to redefine the "pad" on a different machine, but that can be done without unpacking or repacking the entire record.

6.5 Section 3 - Data Description Section.

6.5.1 Data Subsets. "Data subsets" are variously defined in the current BUFR documentation. Conceptually, one subset is a collection of "related meteorological data", quoting from the Manual. Continuing: "For observational data, each subset usually corresponds to one observation", where "observation", in this context, could mean one surface synoptic

observation of a number of specific elements, one radiosonde ascent, one profiler sounding, one satellite derived sounding with radiances perhaps, or the like. No examples of non-observational data subsets are given, but a typical one would be a message consisting of a collection of numerical model forecasts of "soundings" at grid-points or other specific locations. Each forecast sounding (pressure, temperature, wind, relative humidity, whatever, at the many levels of the model) would then be one data subset.

A more precise (if slightly tautological) "operational" definition shows up later on in Regulation 94.5.2: "A data subset shall be defined as the subset of data described by one single application of this collection of descriptors." In this context, the "collection of descriptors" means ALL the descriptors included in Section 3 of the BUFR message. In other words, one pass through the complete collection of descriptors will allow one to decode one data subset from Section 4. One then loops back in the descriptor list for as many times as the data subsets count call for. All of the data, in Section 4, are properly described by repeated use of the same set of descriptors.

This does not imply that the data subsets are themselves identical in format. The use of delayed replication, as in a collection of RAOBs with varying numbers of significant levels, could cause variations in format (octet count) among data subsets. But they are still considered "subsets" in that the same set of descriptors will properly describe each individual set. The use of the delayed replication descriptor is what makes this possible, and is what delayed replication was designed for.

As noted in Chapter 5, certain descriptor operators, from Table C, can be used to redefine reference values, data lengths, scale factors, and add associated fields. There is also a group of descriptors which "remain in effect until superseded by redefinition" (more on them below). By common practice, ALL of these redefinitions or "remain in effect" properties are canceled when one cycles back to reuse a set of descriptors for a new data subset. You wipe the slate clean and start as though it was the first time. This rule is NOT specifically stated in the Manual at present, but presumably will be in the next update.

Of course, data subsets can be identical in format, i.e., have the same number of octets in each subset. This will always be the case if delayed replication is avoided. In this case one can compress the data, as described in Chapter 4, and gain considerable efficiency. Chapter 4, in the interest of avoiding overwhelming detail, doesn't mention that it is perfectly possible to compress data elements to which have been attached associated fields. The catch is that every data element has to have an associated field attached to it for the systematic compression to be possible. This may cut into the efficiency of the compression and should be considered before undertaking such a project.

Even though data subsets may be compressed and, as a result, the individual elements in each data subset are all reordered, the data subset concept still holds. The data subset count must be included in the correct location, and must be correct, of course. It is impossible to decompress a message without that information; and even if the data are not compressed the count is necessary to retrieve all the data subsets in a given message.

A final note about subsets: It is possible, within the BUFR framework, to account for many subsets by the device of placing a replication operator just in front of the set of descriptors that define one subset and have that replication include the count of all the subsets. This in effect reduces the data down to just one subset in that one would no longer cycle back and reuse the complete set of descriptors (now including the replication descriptor). This is NOT a recommended procedure. It is far better to have the subset count "up front", so to speak, in octets 5-6 of Section 3 if for no other reason that it gives the user an indication of how much data he will have to contend with before the decoding gets under way.

6.5.2 Observed or "other data". A brief note: the "other data" flagged in octet 7 has been taken to mean forecast information, such as a collection, from a numerical model, of forecast "soundings" of wind, temperature, humidity, whatever, at the various internal layers or levels of the model, at a collection of grid points or interpolated locations. The time significance qualifier (0 08 021) is used to indicate that the hours associated with each sounding are indeed forecast hours. The initial time of the forecast is given as an unqualified date/time group, and it is in the message prior to the 0 08 021 descriptor.

6.5.3 Data Descriptors. Here is where we shall discuss some of the advanced, tricky, quirky, or special features about descriptors. Perforce, there will be collateral discussions of the data which those descriptors set out to describe. Much of what is discussed here is in the nature of meta-rules about descriptors, in that it deals with the proper interpretation of some special descriptors and interpretation of special combinations of descriptors.

Descriptors, in isolation, are rather straight-forward: one descriptor describes one piece of data, one to one (or in the case of Class D descriptors, one to many). The special rules discussed here go beyond that - some are, in effect, the rules that an application program needs to "know", given that a set of (presumably decoded) data, with associated descriptors, is presented to it. The application program has to "know" the "meaning" of these special descriptors, or patterns of descriptors, to handle the data properly and deliver to the end user what the constructor of the BUFR message intended. Some of the meta-rules are also in the nature of operator descriptors that the BUFR decoding program itself has to "know" in order to reconstruct the original data. Of course, the creator of such BUFR messages has to know and follow the rules as well.

Perhaps all this generalization will come clearer when we deal with specific examples.

6.5.3.1 Descriptors for "Coordinates". The descriptors in Classes 00 through 09 (with 03 and 09 at present reserved for future use) have a special meaning added to them over and above the specific data elements that they describe. They (or the data they represent) "remain in effect until superseded by redefinition". By this is meant that the data in these classes serve as coordinates (in a general sense) for all the following observations. Once you encounter an 0 04 004 (which describes the "hour") one must assume that the hour (a time coordinate) applies to all the following observations, until either another 0 04 004 descriptor is encountered or you reach the end of the data subset.

Obviously the familiar coordinates (two horizontal dimensions - Classes 05 and 06 - a vertical dimension - 07 - and time - 04) are in this subcategory of descriptors, but so are some features that one might not think of as "coordinates", other than in a general sense. Forms of "identification" of the observing platform (block and station number, aircraft tail number, etc.) are "coordinates" in this sense, in that they most certainly apply to all the observations taken from that platform and they "remain in effect until superseded by redefinition". The instrumentation that is used to take the measurements (Class 02) also falls in the same category - it applies to all the actual observations because all the observations were made with that particular instrument. (A lot of the instrumentation class deals with details of radar - there seems a lot more to say about such equipment than, say, a thermometer. But if reporting details about the thermometer [mercury vs. alcohol vs. bimetallic strips, say] became important this information could be added to Class 2 without difficulty.)

A source of confusion can arise by noting that some parameters (height and pressure, for example) appear twice in the Tables: in Class 07 and again in Class 10. Which table descriptor is appropriate depends on the nature of the measurement that involves these parameters. A radiosonde, which measures wind, temperature, and humidity (and geopotential height by calculation) as a function of pressure, would report the pressure values using Class 07 (the vertical coordinate or independent variable) and the other parameters from the non-coordinate classes (10 for geopotential, 11, 12, and 13 for the others). An aircraft radar altimeter, on the other hand, might measure pressure (and use Class 10 to report the value) as a function of height (Class 07).

Yet another kind of "coordinate" is imbedded in Class 8 - Significance Qualifiers. These are a way of reporting various qualitative pieces of information about the (following) data elements, beyond their numeric values, that can be important to the user of the data. A problem of how to "cancel" significance has come up - there are cases where it makes no sense to have a particular kind of significance "remain in effect" for the rest of the message (or to the end of the data subset) but there is no explicit way to cancel it. A convention has been more or less agreed to that sending a "missing" from the appropriate table has the effect of canceling whatever significance was previously established from that table.

Presumably, this convention will become a rule (or footnote) in a future printing of the BUFR manual.

There is an exception to the "remain in effect until redefined" rule: when two identical descriptors, from Classes 04 to 07, are placed back to back, that is to be interpreted as defining a range of coordinates. In this way an area, a volume, a span of time, or all three together, can be defined as needed. If the same descriptor shows up later on in the message, then that appearance does indeed redefine that particular coordinate value. The others still remain in effect.

Unfortunately some coordinate-like information has appeared in a Table outside the Class 00-09 range - it escaped somehow. Class 25 - Processing information, largely dealing (again!) with radar information, contains information that by its nature "remains in effect until superseded". It should be considered as a "coordinate" class and most likely will get such an official designation in the future. This will not involve any changes to the structure of BUFR or the tables, only a change in interpretation, or "meaning", of the data elements.

There is not much a general BUFR decoder program can do with this "coordinate " information, other than decode it and pass the information on to some follow-on applications program. As noted in the introduction to this sub-section, it is up to the applications program (or the human reading a decoded message) to supply the interpretation and the meaning of what is there, and then to act accordingly. Some of the interpretation is straightforward, almost second nature. "Obviously" the station identification applies to the following observations made at that station; "obviously" this pressure level is where the RAOB measured the wind and temperature; perhaps not so obvious is the fact that two consecutive azimuth values define a sector in which a hurricane is located. Making the "obvious" explicit with rules, regulations, and footnotes is part of what BUFR is all about. The developers of BUFR made every effort to EXCLUDE as much "self-evident" information as possible and instead require that "meaning" be specified by definite rules - that is, in part, what makes the system so powerful. [End of second commercial!]

6.5.3.2 Replication, Increments and "Run-length encoding". As described in Chapter 3, replication (a descriptor with F=1) is pretty straightforward. Even delayed replication is no real problem (except to someone writing a program to do it correctly). In either case, you just replicate the following X descriptors Y times ("Y" can be either part of the descriptor or found in the data section) and that is it. This allows you to encode and describe a potentially very large amount of data with relatively few descriptors. Very powerful feature.

The only slightly tricky matter is to keep mind that the 0 31 YYY descriptor that follows the delayed (Y=0) replication descriptor is not included in the count of descriptors to be replicated, the XX part of 1 XX YYY. Indeed the descriptors of Class 31 hold a unique position in BUFR. With one (partial) exception, they are never used in isolation, but always in

conjunction with some other descriptor in order to "complete" the latter's function. The exception is 0 31 021 - it can be used alone to redefine the meaning of a previously established associated field. Class 31 descriptors are not included in the replication counts for replication descriptors (nor are they replicated), and their characteristics are not altered by any of the operator descriptors in Table C, even those that change a characteristics of every (other) Table B descriptor. They are "Teflon" descriptors: they stick to other descriptors but nothing sticks to them.

A rather ingenious "extension" to the delayed replication concept has come into use recently. This is one of those "unrecognized possibilities" of BUFR mentioned previously. The idea is simple: set up delayed replication but have the replication count (in the data section) be equal to zero. By a simple extension of the rules, this clearly means that the "following X descriptors shall be replicated zero times", that is, they don't get used at all, they should be skipped over - there is nothing in the data section corresponding to them. This is quite useful in that it allows one to set up a standard or all inclusive set of descriptors for a variety of observation types but then tailor the use of the descriptors, by setting the replication count to 1 or 0, to fit the actual data in hand. It is considerably more efficient than filling in the "missing" data (all 11111 bits) in the locations in the data section where there is no real observation. A particular example of this is in "vertical soundings", whether generated by RAOBs, satellites, profilers, dropsondes, etc. They all share a basic common structure but some lack whole classes of data - satellite soundings have no winds, for example. The use of "zero count replication" allows one to set up a single set of descriptors for all of these observations with a net saving of space over either setting a lot of "missings" in the data or maintaining a library of different sounding descriptor sets.

The current descriptors allow zero count replication without any changes in current tables. However, to save a little more space, the NMC (Washington) people have defined a 0 31 000 descriptor with a 1-bit data length. This allows a replication count of 1 or 0, all that is needed. This is not yet officially recognized (even though it is within the international portion of the table), but there seems little reason to doubt that it soon will be. It is a very useful idea.

When we turn to the few descriptors that define increments, and in particular discuss the use of increments in conjunction with replication, things get a little complex. The rules get quite precise and have to be adhered to closely.

Increments by themselves are not so bad. One first establishes the value of a coordinate that is capable of being incremented. Normally, that coordinate value would "remain in effect until superseded" by the appearance of the same descriptor with a new data value. But the appearance of a descriptor for an increment associated with that coordinate will also change the value of the coordinate by the amount found in the data section. The increment descriptor must be in the same class as the data to be incremented and must have the same units. In the current BUFR tables there is no built-in way to associate an increment uniquely with the descriptor/value that is capable of being incremented. This is unfortunate

as it means the decoder program must have special rules encoded for each increment descriptor; it would be better to devise a general rule to associate increments with the thing (or things) to be incremented. This is a project for the future.

A sample is the best way to indicate the descriptor sequence when increments and replication are combined:

Descriptor	Interpretation
0 04 004	Sets the value of the hour at one increment LESS than the "starting" value.
. dddd dddd	assorted data may be placed here without influencing the replication to come
. 0 04 014	sets the value of the increment in hours and increments the hour
1 XX 000	set up (delayed) replication of "next" XX descriptors
0 31 001	replication count (not included in the span of replication XX)
. . .	XX descriptors to be replicated

Regulation 94.5.4.3 says that when the increment descriptor just proceeds the replication operator, as in this example, the incrementing action takes place right along with the replication. Every time the descriptors are replicated the hour (in the example) gets incremented, too. Note also, that the hour gets incremented right away, before the first pass through the XX descriptors. That's why the initial hour value (0 04 004) was given a value one increment's worth less than the hour value needed for the first iteration.

There is a refinement to this: it is legitimate to place Table C Operator Descriptors between the increment descriptor and the associated replication operator without altering the rule that the incrementing is associated with the replication. This is to allow for (temporary) redefinition of the data width, scale, whatever, of the descriptors within the XX span of replication (and following unless the changes are canceled), if necessary. The class C descriptors cannot be placed after the replication count descriptor as they would then be subject to the replication which might not work very well, nor can the class C descriptors be placed prior to the increment descriptor itself as that means the increment descriptor would have its characteristics changed, also not a good thing. Hence the refinement to the rule. (Don't forget the other rule, that Class 31 descriptors are not subject to change by Table C descriptors.)

Another feature of replication is "run length encoding". This is enabled by replication followed by the 0 31 011 (or 0 31 012) descriptor. Basically all it says is that in addition to replicating the descriptors a number of times, the data elements

present in the data (as described by the set of descriptor to be replicated) should be replicated as well. This is useful, of course, when the original data, as it exists prior to BUFR encoding, contains long runs of identical values, or long runs of identical sets of data elements. This is a familiar and very straightforward form of data compression that can greatly increase the efficiency of data representation in special cases. Of course, the run length encoding replication can be coupled with incrementing of a coordinate; indeed it most likely would be as there is commonly a need to specify the locations of the string of replicated values.

6.5.3.3 The Associated Field. Associated fields are generally for the purpose of "saying something" extra about the particular data element with which they are associated. The most common use is in the arena of "quality control", where some sort of "confidence" indication is given. Other applications are possible and can be established by additions to Code Table 0 31 021.

Creating (or dealing with) an associated field in a message is a two step process. The first is to establish the field and set the number of bits that will precede all the data elements following the appearance of the associated field operator (2 04 YYY). YYY is that number. If 255 bits is not enough (good grief, why?) you can keep adding more bits by repeating the operator. You can also generate compound associated fields by repeating the operator if what you have to "say" about the data elements is complicated.

The second step is to define the meaning of those bits, i.e., how they are to be interpreted by a user of the data. This is done by immediately following each 2 04 YYY descriptor with the usual Class 31 descriptor, 0 31 021, which, by reference to the Code table 0 31 021, establishes that meaning. A little care is required here. Code Table 0 31 021 gives a (small) number of significance code figures (all taking up 6 bits in the data) for different size associated fields; obviously one must be consistent in setting an associated field length and identifying the meaning of the bits in the field.

Once an associated field is established, those extra bits must be (are assumed to be) prefixed to every following data element, until the associated field is canceled. If the quality information has no meaning for some of those following elements, but the field is still there, there is at present no explicit way to indicate "no meaning" within the currently defined meanings. One must either redefine the meaning of the associated field in its entirety (by including 0 31 021 in the message with a data value of 63 - "missing value") or remove the associated field bits by the "cancel" operator: 2 04 000. If multiple or compound associated fields have been defined, each must be canceled separately.

6.5.3.4 Changing Descriptors "On the Fly". A set of descriptors are defined in Class 00 which are used to describe descriptors. These have not had much international (or non-local) use to the best of my knowledge but their purpose, of

course, is to send new international (or local) descriptors to interested parties for use prior to some official publication. But another "new possibility" has been suggested, one that would seem to have considerable potential value. This "new possibility" is not defined in the current BUFR specifications and, as will be obvious, would require a new Edition number for BUFR as it would require changes in the logic of a decoding program.

The suggestion is simple: it should be considered legitimate to send any descriptor, or collection of descriptors (new or currently defined, international or local), imbedded in a message which otherwise contains data. Then the new descriptor(s), or the redefined old one(s), may then be actually used in the remainder of that message/record. This affords a method of introducing new data on the fly, so to speak, or to change specific descriptor characteristics more selectively than can be done at present with Table C (operator) descriptors. Implementing this would, perforce, require that the decoding program recognize the new descriptor and then either add it to some internal table or use it to alter portions of existing tables. Either option would require new rules to be promulgated and old decoders to be altered. It doesn't seem to be a very complicated modification. This temporary change to a descriptor would only hold for the one record or data subset in that record in which the change is introduced. The next BUFR record would be assumed to contain only "standard" (i.e., published) descriptors until such time as more new ones are introduced.

6.5.3.5 BUFR Records in Archives. A simple extension of the "new possibility" rule in the previous section makes it possible to alleviate a big concern about using BUFR records in long-term archives, that is, the necessity to retain BUFR Tables through a number of possible versions for an indefinite time span. The suggestion again is simple and rather obvious. In any file of (presumably many) BUFR records, the first such BUFR record should contain nothing but a collection of all the descriptors that will be used in all the other records in the file. Such a record would have a Table A data category value of 11. The "new rule", then, would be that the descriptors in the first record should be used for decoding all the many records in the file. Individual records could also have redefinitions of descriptors, as above, but they would hold for only the one record or data subset in that record. This is really not a rule about the structure of BUFR per se, but is more of a suggestion for good data management where BUFR records and files are involved. Presumably such BUFR archive files would remain intact and only be exchanged in toto.

This archive suggestion would not involve any changes to BUFR itself (and hence no change to the Edition number) if the construction of Tables B, C and D, based on what is found in the first Table A = 11 record, was done externally to the decoding process. If the temporary change/addition to a descriptor was allowed that would introduce a new Edition to BUFR.

Chapter 7

Use of Binary Representation at ECMWF

J.K. Gibson

7.1 Introduction. The principle function of the European Centre for Medium-Range Weather Forecasts (ECMWF) is to produce daily a medium-range (up to 10 day) forecast, and to distribute products to its Member States. A secondary role is to maintain an archive of Meteorological Data, mainly for the support of internal research, but also for the benefit of the Member States.

Since the forecasts are global in domain, and since the analyses on which the forecasts are based use all available observational data, the ECMWF archive is designed to meet the major need, which is "case study" type retrieval of all data relating to one analysis and/or forecast. In fact, the entities stored within the archive can be addressed at the level of single observations, and single analysis of forecast "fields" (i.e. 1 parameter at one level for all horizontal locations at one point in time).

Data within the applications environment are thus retained either as whole observations (or sets of observations in the case of satellite data), or as fields. The same policy is followed for archived data. This enables applications to use on-line or archive data without modification. The WMO representations BUFR and GRIB are used for observations and fields respectively. These forms, being machine independent, enable data to be transported across the full range of mainframes, servers, and workstations which currently comprise the ECMWF computational facilities.

7.2 Operational Data Management. ECMWF receives observational data from the WMO GTS via Bracknell and Offenbach. Meteorological messages from the GTS are passed as files using a file transfer protocol. Data are acquired, and stored message by message in a structured message data base.

A pre-processing system, driven by the incoming messages, converts the observations into BUFR, and received analysis and forecasts products into GRIB. The BUFR observations data are stored in a reports data base (RDB), the GRIB products in a fields data base (FDB). The data base structures used allow access either to individual BUFR or GRIB entities. From this point on, all applications use or generate data in BUFR and GRIB.

For each data assimilation cycle, observational data are extracted from the RDB, and appropriate first guess and other

fields from the FDB. The resulting analysis and new first guess fields are written to the FDB. The forecast from the 12:00 UTC analysis is continued out to 10 days, and its fields written to the FDB.

The analysis system compares observational data against the first guess, against uninitialised and initialised analysis values, and performs a number of validity checks. Results from the (often referred to as "feed-back" or "analysis statistics") are represented in BUFR, and subsequently used by a number of monitoring applications.

Products from analyses and forecasts are generated to the individual requirements of each of the Member States. Most products are fields, generated in GRIB; some, however, represent time series values of specific parameters at single points throughout the forecast, and are generated in BUFR.

All observations for each day are extracted two days later (allowing for complete reception), sorted, and added to the archive. A similar strategy is followed for the feed-back from the analysis. The global fields containing the analysis and forecast results are also archived.

The archive retrieval mechanism has been developed to make the data residence transparent to the user. If a retrieve request can be satisfied from on-line data (FDB or RDB) this is done. If not, the off-line archives can be accessed. It is thus possible to incorporate calls to the archive retrieval as the standard interface to both on-line and off-line data, enabling operational applications to run unchanged on archive data.

7.3 Use of BUFR. Observational data, when received, is first converted to BUFR. This is achieved through a set of pre-processes which currently run on a VAX cluster.

Although experimentation has been done to investigate the effectiveness of a relational data base, currently the operational system uses a data base system developed in-house, using the VMS indexed sequential file structure. Each BUFR entity is uniquely identified by a key. This key, together with some additional housekeeping information (such as time of receipt, time of pre-processing, message origin, etc.) are retained in the optional Section 2 of BUFR. Information from this key is used for sorting and post-processing within the archive retrieval system.

Whenever possible, the BUFR used to represent observational data conforms strictly to the WMO standard. The representation of analysis feed-back or statistics data in BUFR is a somewhat more recent development, and uses features of BUFR currently approved for experimental use. ECMWF experimentation using extensions has illustrated that the basic principles involved are effective; it has also revealed areas where some improvement is possible before the

experimental extensions are incorporated within the full BUFR definition. The concept of trying out new extensions to BUFR in a full processing environment is extremely beneficial. It enables one or two centres to try out new concepts, determine their validity, and possibly recommend improvements before they become part of the full BUFR specification.

All of the observational data within ECMWF's archive from 1980 onwards have now been converted to BUFR.

In addition to the operational use of BUFR, the observational archive is used extensively for research into better data assimilation methods, and for the verification of forecasts (operational and research) against observations.

Currently as a build up phase towards a project to re-analyse 15 years of data the ECMWF archives are being enhanced. Data are being added from cloud cleared radiances, from COADS, from the Australian archive of PAOB data, and to resolve other known deficiencies. In addition all FGGE and ALPEX II-B data are being converted to BUFR and added to the archives. The re-analysis project will result in BUFR feed-back data for the full 15 years, 1979 through 1993.

7.4 Use of GRIB. GRIB has been in use at ECMWF for about 8 years, and all results of forecasts and analyses (operational and research) are generated and archived in this form.

Most Member States products are generated in GRIB and distributed in this form. This avoids complications due to the many different types of computers in use at Member States.

In recent years the demand for ECMWF products world-wide in non-real time has grown considerably. Retaining the data in a standard WMO representation form has enabled such demand to be met with a minimum of re-processing. Since GRIB handling is well understood throughout the world it has also ensured a minimum of follow-up action when data have been delivered, as most recipients have little difficulty in handling the data.

Recently the ECMWF archive has been re-processed to generate GRIB time series and monthly means.

GRIB products from the archive are used extensively for the initial conditions for research experiments, and for the verification of research experimental forecasts.

An extensive set of results from the re-analysis project will be archived in GRIB, including monthly and seasonal means, and a considerable number of additional statistics.

7.5 Concluding Remarks. Use of standard binary representation has brought the following benefits:

- machine independent data
- efficient data representation
- standard interfaces to applications
- provision of data to external users in an acceptable form
- simplified and efficient data management.

To date there are no known negative aspects.

APPENDIX A

REFERENCES

1. Soderman, D. and Gibson, J.K. "The Specification for FM 94 BUFR". FM 94 BUFR Collected Papers and Specification. ECMWF, February 1988.
2. Stackpole, J. "Binary Universal Form for Data Representation (WMO Code FM 94 BUFR)". FM 94 BUFR Collected Papers and Specification. ECMWF, February 1988.
3. World Meteorological Organization Manual on Codes, Volume 1, International Codes, Part A - Alphanumeric Codes. 1988 Edition, Suppl. No. 2 (VII.1991)
4. World Meteorological Organization. Manual on Codes, Volume 1, International Codes, Part B - Binary Codes. 1988 Edition, Suppl. No. 3 (VIII.1991)

