# CHAPTER 6

## QUIRKS, ADVANCED FEATURES, AND SPECIAL USES OF BUFR

(contributed by: 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 all eventually find its way into the published specifications.

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 into the existing system, as though they were implicitly present from the beginning. At other times, 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 and these 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 and this, in turn, leads to complexity. There is no free lunch.

As an organizing structure, each section of a BUFR message/record will be dealt with in its 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. This difference 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 at the end of par. 1.2.1. These types of 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 B, 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 B, Class 31. This is 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 apt to happen - more likely there will be clarifications to the regulations or notes that will serve to make the rules more precise in cases that currently seem ambiguous. 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, as specified in the Manual on the GTS, WMO Publication 386, 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 new GTS specification for breaking up very large bulletins, using the new BBB parameter in the WMO Abbreviated Heading, has recently been promulgated. 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. It is one of the major strengths of BUFR that any scientific discipline interested in transmitting, storing, or even data-basing its unique information may 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 (SGDR&C). International notification of such additions is accomplished by the World Weather Watch (WWW)

Operational Newsletter. 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. 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 and that descriptors will neither be deleted nor changed. Thus, existing messages and decoding tables will not be affected as long as they have no need to make use of the new data descriptors. Changes to the tables which involve only additions do not require that the version number of the tables be changed. Also, changes which are in the nature of "trivial" corrections (typographical errors, more precise definitions of terms, etc.) do not engender new version numbers. The SGDR&C will define what is "trivial" and what is not. At present, the tables stand at Version 2.

The SGDR&C 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, wholesale reorganization of the tables, or the possible elimination of old and unused descriptors. The latter two steps 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 editions of BUFR, or new versions of the tables, upon approval of the CBS. Because both the BUFR edition number and the version number of the tables are part of the BUFR message, it is only a programming task for a decoding program to note the BUFR edition number of a message and the version number for the tables 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 usually be sent out internationally (but see the discussion of descriptor 2 06 YYY in paragraph 5.5 for the way to handle any 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 and sub-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**. The method of specifying the number of the originating center has been changed from what is described in

the (current) Manual on Codes (Supplement 3, 1991). Here is a little historical background as to how things have evolved. GRIB (FM 92) was developed first and adopted a pre-existing WMO table of meteorological centers for "originating 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 that the GRIB table could accommodate. Thus, in BUFR, two octets were set aside for numerical specification of those locations, where GRIB used but one. A proposal was developed to enumerate those additional locations based upon International Civil Aviation Organization (ICAO) Location Indicators and this was published in the 1991 supplement as part of the BUFR specifications. Since then, however, it was realized that confusion and inconsistencies could result from separate GRIB and BUFR originating center tables and a recent proposal was accepted to construct tables that were common to GRIB, BUFR and any other WMO code. To do this it was, in turn, necessary to drop the ICAO numbering system from BUFR. Fortunately, the two tables have not, up to now, developed any inconsistencies and the "ICAO numbers" are in very limited use. It was concluded that this change could be done without requiring a new edition for BUFR.

The resulting system is simply that octet 6 of Section 1 - Identification Section is used to identify the national (or international) originating centers, using the same common table as is in use for GRIB. This table will be coordinated and maintained by the WMO and published as part of the codes manual. Any national sub-center numbers that may be required are to be generated by the national center in question and that number is to be placed in octet 5. The WMO has expressed a willingness to publish sub-center identification tables as supplied by the national centers.

**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 later in this chapter. If it is present, the flag in octet 8 of Section 1 must be set, of course.

**6.3.5 BUFR Message Sub-Type.** This is purely a local option. As an example, beginning on the following page are listed the sub-types currently in use at the National Meteorological Center, Washington.

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 |

The above 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.

**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), Section 2 - 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 | ddd1 | vvv1 |
| record (ignore blank lines) | ddd2 | vvv2 |
| | ddd3 | vvv3 |
| end of "real" data subset ------> | ddd4 | vvv4 |
| | | |
| Delayed rep. of two | 1 02 000 | - |
| descriptors n times; | | |
| n is the number of bits in | 0 31 001 | n |
| 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 to the padding problem, of course, is 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 must not be compressed. A common use of a meteorological 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 record.

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 calls 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 to 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 of Section 3 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.

"Other data" need not be limited to forecasts, of course. Statistical, climatological, quality control information, etc. would all fall under the general category of "not observations". This lack of specificity is not of very great concern as the descriptors in the

body of the message take care of the precise definition of just what information is in the BUFR record.

**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 sub-category 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 upon 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 even if the original coordinates defined a range. The others still remain in effect for any subsequent data.

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. A very powerful feature.

The only slightly tricky matter is to keep in mind that the 0 31 YYY descriptor that follows the delayed (Y=0) replication descriptor (1 XX 000) 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 more 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. (Should be set to one increment LESS than the starting" value.) |
| . | |
| . | |
| . | |
| dddd | assorted data may be placed here |
| dddd | 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 replicat-

ing the descriptors a number of times, the data elements present in the data (as described by the set of descriptors 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 that 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 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. 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.