Description and Format of DX BUFR Tables

(NOTE: this document is intended to be read in tandem with a sample BUFR tables file, which can itself be viewed within a separate window by clicking here.)

As noted during the discussion of subroutine OPENBF, every BUFR file that is presented to the BUFRLIB software, either for input (reading/decoding) or output (writing/encoding) purposes, must have DX BUFR tables associated with it, unless the 'SEC3' decoding option is specified during the call to OPENBF. For all other cases, DX table information must be pre-defined and made available to the software via call argument LUNDX during the call to OPENBF. In the case of an existing BUFR file, the DX tables information may be embedded within the first few BUFR messages of the file itself. Otherwise, a separate ASCII text file containing the necessary DX tables information must be supplied, and this document describes the required format and content for such a file. It is extremely important that any such file not only be syntactically correct but also complete, in the sense that all necessary "mnemonics" must exist and be fully-defined.

First, let's define what we mean by a mnemonic. In short, a mnemonic is simply a descriptive, alphanumeric name for an data value. In the context of the BUFRLIB software, there are "Table A mnemonics", which refer to particular data subset (i.e. report) types, "Table B mnemonics", which refer directly to basic data values, and "Table D mnemonics", which are sequences composed of one or more Table B (or other Table D!) mnemonics and which are themselves normally direct constituents of a particular Table A mnemonic. In other words, at the highest level, we have a Table A mnemonic which completely describes a type of data subset (e.g. rawinsonde, wind profiler, etc.), and this Table A mnemonic is defined as a sequence of one or more Table B or Table D mnemonics, where each Table D mnemonic is likewise itself defined as a sequence of one or more Table B or Table D mnemonics, and so on until the entire data subset can be equivalently described as a sequence of one or more Table B mnemonics which, again, themselves correspond to basic data values (e.g. pressure, temperature, humidity, etc.). In this way, the entire sequence of data values that constitute a particular type of data subset is fully and unambiguously defined, both for purposes of input (reading/decoding) or output (writing/encoding) of reports corresponding to that particular type of data subset.

However, it is also important to understand what mnemonics are not. Specifically, mnemonics never themselves appear within actual BUFR messages that are read or written by the BUFRLIB software; rather, their only purpose in life is to make it easier for users to interact with the software by providing descriptive names to represent individual data values, as opposed to having to keep track of the corresponding FXY numbers (described below), which are much less intuitive but which nevertheless are the prescribed method within the BUFR code form for referencing of individual data values (and which therefore are what is actually read and written by the software!).

As we begin our actual discussion of BUFR tables files, let's start with an overview, by noting that a BUFR tables file consists of three distinct sections, each of which contains one or more lines of 80 characters in length, and where a "*" as the first character of a line indicates that that entire line is a comment. In the first section, all Table A, B and D mnemonics that are to be defined within the file are initially declared, assigned a unique FXY number, and given a short, free-form text description. Then, in the second section, all previously-declared Table A and Table D mnemonics are actually defined as a sequence of one or more Table B (or other Table D!) mnemonics. Finally, in the third section, all previously-declared Table B mnemonics are defined in terms of their scale factor, reference value, bit width, and units.

Now, as we delve into the details of each of the three sections, we will constantly refer back to our sample BUFR tables file in order to better illustrate the concepts that are discussed.

Section 1

As previously mentioned, the first section of a BUFR tables file is where all Table A, B and D mnemonics are initially declared, assigned a unique FXY number, and given a short free-form text description. Mnemonics may contain any combination of uppercase letters and numbers (or, in certain special cases, a "." character!), up to a maximum total of 8 characters in length. A mnemonic may be declared only once, and each one must correspond to a unique FXY number, which itself consists of 6

characters, and where the first character (i.e. the "F" component) is an "A" if the mnemonic is being declared as a Table A mnemonic, "3" if the mnemonic is being declared as a Table D mnemonic, and "0" if the mnemonic is being declared as a Table B mnemonic. Otherwise, the remainder of the FXY number must be all digits, with the next 2 characters (i.e. the "X" component) as a number between 00 and 63, and the final 3 characters (i.e. the "Y" component) as a number between 001 and 255. Readers who are more familiar with BUFR will immediately recognize these F, X, and Y values as those that are defined within the official documentation of the BUFR code form; therefore, by international convention, a mnemonic should not be given an X value between 00 and 47 along with a Y value between 001 and 191 unless that mnemonic, when subsequently defined, corresponds exactly to the BUFR descriptor having that same FXY number within the official, internationally-coordinated WMO BUFR tables. For example, in our sample BUFR tables file, mnemonic "WMOB" is declared with an FXY number of 001001; therefore, it has the exact same text description (i.e. "WMO BLOCK NUMBER") and, when later defined within the last section of the file, the exact same scale factor, reference value, bit width, and units as for FXY number 001001 within the official BUFR tables. This concept should be somewhat intuitive, but it is obviously very important when the BUFRLIB software is to be used to write BUFR messages that may potentially be read by other users in other organizations around the world.

In looking further at our sample BUFR tables file, we see that the lines within the first section each contain a "|" character in columns 1, 12, 21, and 80. Mnemonics are declared, and are left-justified, in columns 3-10, corresponding FXY numbers are assigned in columns 14-19, and the corresponding text description begins in column 23. All of the Table A mnemonics are declared first, followed by all of the Table D mnemonics, followed by all of the Table B mnemonics. Within each set, it is generally a good idea for human-readability purposes to list the mnemonics in ascending order with respect to their FXY number, although this is by no means a requirement within the BUFRLIB software itself. Likewise, human-readability can usually also be improved by the judicious use of one or more "separator" lines containing the required "|" character in columns 1, 12, 21, and 80 but without any actual mnemonic declaration; however, again, the use of such "separator" lines is not required by the software. In fact, the software will simply continue reading lines of the file, one at a time, and looking for new mnemonic declarations, until it reaches a line which does not contain a "|" character in each of columns 1, 12, 21, and 80, at which point it then knows that the first section of the tables file has ended.

We mentioned earlier that mnemonics only exist in order to facilitate user interaction with the BUFRLIB software and that, therefore, mnemonics should be as intuitive as possible. We now need to amend that statement slightly, because certain Table A mnemonics do have a special additional function. Specifically, if a Table A mnemonic consists of 8 characters (i.e. the maximum) and if characters 3 through 8 are all digits, then the mnemonic is also used by the software to set the data category and local subcategory within Section 1 of each BUFR message when writing/encoding data subsets corresponding to that mnemonic. In such cases, characters 3 through 5 define the category, and characters 6 through 8 define the subcategory. Therefore, in referring again to our sample BUFR table where we have defined three different Table A mnemonics, we have also indicated that, e.g. when we use the software to write/encode data subsets according to the Table A mnemonic "NC002007" (i.e. wind profiler), we want all BUFR messages which contain such data subsets to be encoded as category 2 and local subcategory 7 within Section 1 of the message!

Incidentally, even if a Table A mnemonic does not meet the above criteria, BUFR message category and local subcategory values will still be set by the software when writing/encoding BUFR data subsets corresponding to that Table A mnemonic. However, in such cases, the category value will be set to the "Y" component (i.e. last 3 digits) of the FXY number corresponding to the mnemonic, and the subcategory value will simply be set to 0. Therefore, it is recommended to use the previous, more-explicit approach when assigning a Table A mnemonic for a data subset to be output, since this approach provides for greater control over the category and subcategory values that will be encoded into Section 1 of the resultant BUFR message. We should also take this opportunity to point out that, when the FXY number corresponding to a Table A mnemonic is actually encoded into a BUFR message, a "3" is actually encoded in place of the "A" which is used in the tables file. Put another way, the "A" that appears within the FXY number corresponding to each Table A mnemonic within the tables file is only there so that such mnemonics can be easily distinguished from Table D mnemonics by the software.

Section 2

Now, let's move on to the second section of a BUFR tables file. As already stated, this section is used to define, for each Table A and Table D mnemonic that was previously declared in the first section, the sequence of Table B (and possibly other Table D!) mnemonics which constitutes that mnemonic. The format for this section is a "|" character in columns 1, 12, and 80, with the mnemonic that is being defined listed in columns 3-10 (left-justified), and the sequence of constituent mnemonics beginning in column 14, each one separated from the others by one or more blank characters. For longer sequences, multiple successive lines may be used in a continuation fashion by repeating, within columns 3-10 of each continuation line, the mnemonic being defined. For example, in our sample BUFR tables file, the Table D mnemonic "MRPSC0" is defined as consisting of the sequence "YEAR", "MNTH", "DAYS", "HOUR", "MINU", "RPID", "MRPIDS", "CLON", "CLAT", "SELV", and "CORN", where "MRPIDS" is itself a Table D mnemonic which is therefore itself defined in a similar manner elsewhere within the section. As was the case with the first section, "separator" lines may be employed within this section in order to improve human-readability, as long as they contain the "|" character that is required to be in columns 1, 12, and 80 for all non-comment lines within this section, and the BUFRLIB software will continue reading lines of the file as though they are part of the second section until it encounters one that does not adhere to this format.

At this point, most readers who have taken at least a cursory glance at the sample BUFR tables file will have no doubt begun to wonder about all of the additional punctuation characters and symbols included within the sequence definitions of the second section. It is now time to address these concerns by stating that these are replication indicators for the mnemonic(s) in question:

< > indicates that the enclosed mnemonic is replicated using 1-bit delayed replication (either 0 or 1 replications)
{ } indicates that the enclosed mnemonic is replicated using 8-bit delayed replication (between 0 and 255 replications)
( ) indicates that the enclosed mnemonic is replicated using 16-bit delayed replication (between 0 and 65535 replications)
" "n indicates that the enclosed mnemonic is replicated using regular (non-delayed) replication, with a fixed replication factor of n

Examples of most of these cases are shown within the sample BUFR tables file, and, through successive application, can lead to the definition of some rather interesting data structures! For example, the Table A mnemonic "NC002001", which defines the layout of a data subset of the type "rawinsonde - fixed land", consists of the following sequence:

"UARTM", followed by
between 0 and 255 replications of "RCPTIM", followed by
between 0 and 255 replications of "BID", followed by
"UASID", followed by
between 0 and 255 replications of "UARID", followed by
between 0 and 255 replications of "UARLV", followed by
either 0 or 1 replications of "UASDG", followed by
between 0 and 255 replications of "UARDCS", followed by
between 0 and 255 replications of "RAWRPT", followed by
between 0 and 255 replications of "UACLD", followed by
either 0 or 1 replications of "UAADF", followed by
"WMOB", followed by
"WMOS", followed by
"WMOR"

where, e.g., the constitutent Table D mnemonic "UARLV" itself consists of the following sequence:

"VSIG", followed by
"QMPR", followed by
"PRLC", followed by
"QMGP", followed by

either 0 or 1 replications of "UAGP07", followed by
either 0 or 1 replications of "UAGP10", followed by
either 0 or 1 replications of "UATMP", followed by
either 0 or 1 replications of "UAWND", followed by
either 0 or 1 replications of "UAWSH"

and where, in turn, "UAGP07", "UAGP10", "UATMP", etc. are also Table D mnemonics which can themselves be further resolved. So we can even nest certain replication sequences inside of other replication sequences, and, further, via the judicious use of the < > indicator, even turn on/off entire sequences of data values simply and efficiently. An example of this is the "UAWSH" (i.e. "RADIOSONDE WIND SHEAR DATA") sequence, whose constituent data values are only ever present in a rawinsonde report when a level of maximum wind is being reported (and, even then, not always!). In this case, enclosing the entire sequence within a < > indicator allows the lack of such data within a report level to be noted by the use of a single bit set to "0" (i.e. 0 replications), rather than having to store the appropriate "missing" value for each constituent data value. Over the course of many data levels within many data subsets within a single BUFR message, this can add up to significant encoding efficiency, and, in turn, the use of less required storage space per BUFR message. So, in summary, the judicious use of replication can even lead to more efficient data storage for certain types of data.

Going back to the sample BUFR tables file, notice how several of the Table D mnemonics such as "RCPTIM" and "BID" are used within both the "NC001003" and "NC002001" data subset types. This brings up a good point; namely, that by logically grouping certain Table B mnemonics together within carefully-constructed Table D sequence mnemonics, such mnemonics can be easily and efficiently re-used within different Table A mnemonic definitions within the same BUFR tables file. In fact, this would be a good time to also point out that, when using the BUFRLIB software, Table D sequence mnemonics are the only types of mnemonics upon which any type of replication may be directly performed. Thus, in particular, if we wish to effect the replication of a single, particular Table B mnemonic, then we must do so by defining a Table D sequence mnemonic whose only constituent is that particular Table B mnemonic and then replicating the sequence mnemonic. For a specific example of such a situation, take a look at the definition of "RAWRPT" within the sample BUFR tables file.

Before we end our discussion on the second section of our sample BUFR tables file, there are a couple of other special situations that we need to explain in further detail!

First, notice how a 201YYY indicator precedes each occurrence of "ACAV" within the definition of the Table D sequence mnemonic "OBSEQ" as well as each occurrence of "HINC" within the definition of the Table A mnemonic "NC002007". This indicator is called an operator, and readers more familiar with the details of the BUFR code form will no doubt recognize it from Table C of the official, internationally-coordinated BUFR tables. In short, the effect of this operator is that, for each Table B mnemonic which follows it within the current sequence, and continuing up until the point in the sequence where a corresponding 201000 operator is reached (and which turns off the effect), (YYY - 128) bits should be added to the bit width that is otherwise defined for that Table B mnemonic within the third section of the BUFR tables file, so that the net effect is to change the number of bits occupied by the data value corresponding to that mnemonic within the overall data subset. Thus, for example, the sequence:

201132 HINC 201000

indicates that (132 - 128) = 4 bits should be added to the data width that was defined for mnemonic HINC within the third section of the BUFR tables file, and, therefore, that for this occurrence of that mnemonic within the overall data subset, the corresponding data value will occupy (12 + 4) = 16 bits.

Other than 201YYY, the BUFRLIB software also supports the similar use of the 202YYY (change scale), 204YYY (add associated field), 205YYY (add character data), 206YYY (define data width for local descriptor), 207YYY (increase scale, reference value and data width) and 208YYY (change data width for CCITT IA5 descriptor) operators from BUFR Table C.

Finally, take a look at the definitions of the Table D sequence mnemonics "TMPSQ3", "WNDSQ2", and "PCPSQ3"; in particular,

notice that, within these definitions, there are references to several mnemonics such as ".DTHMITM" and ".DTHMXGS" which were not previously-declared within the first section of the table. At first glance, this seems to contradict everything that we previously said about the need to initially declare all mnemonics within the first section; however, upon closer inspection, the reader will notice that there do exist, within the first section, declarations for mnemonics ".DTH...." and ".DTH....". So, what exactly is going on here? The answer is that each of these is a special mnemonic known as a following-value mnemonic, meaning that, when it is used within a sequence definition, it implies a special relationship with the mnemonic that immediately follows it within the sequence. In fact, this relationship is so special that, when a following-value mnemonic is used within a sequence definition, the "...." portion of the mnemonic is replaced with the mnemonic that immediately follows it! For example, when ".DTH...." is used within the definition of the Table D sequence mnemonic "TMPSQ3", it appears as ".DTHMXTM" and ".DTHMITM" because it appears immediately before, respectively, the mnemonics "MXTM" and "MITM". However, when it appears within the definition of "PCPSQ3", it appears as ".DTHTOPC" since it immediately precedes "TOPC" within that sequence! To be precise, a following-value mnemonic is declared with a "." as the first character, followed by no more than 3 alphanumeric characters as an identifier, followed by 4 more "." characters which must then be replaced with the mnemonic that immediately follows it whenever and wherever it is used within a sequence definition. This is important, because the BUFRLIB software will actually check that the immediately-following mnemonic matches the last 4 characters of the following-value mnemonic and will diagnose an error if it does not.

In general, the "following-value" attribute is useful because it allows the same mnemonic to be used repeatedly within the same overall Table A data subset definition in a very intuitive fashion and yet, since each occurrence retains its own unique identification (e.g. ".DTHMXTM", ".DTHTOPC", etc.), then each one can still be individually accessed independent of the others via subroutine UFBINT. An alternative would be to declare a regular mnemonic such as "DTHRFV" instead of ".DTH...." within the first section of the tables file and then use that mnemonic in all of the same places within the same Table A data subset definition, but then we would have to use subroutine UFBREP to access all such values simultaneously (even if we weren't interested in all of them!), and we would also lose the intuitiveness provided by having available, within the mnemonic itself, the name of the mnemonic to which the corresponding value applies.

Section 3

It is now time to move on to the third and final section of a BUFR tables file. As we mentioned earlier, this section is used to define the scale factor, reference value, data width, and units for all of the Table B mnemonics that were previously declared in the first section. In particular, the reader may recall that the units definition for each Table B mnemonic in turn determines how data values corresponding to that mnemonic are read/written from/to the REAL*8 array R8ARR within the BUFRLIB subroutines UFBINT, UFBREP and UFBSEQ.

In looking again at our sample BUFR tables file, we see that the format for the third section of such a file is to have our same old, familiar "|" delimiter in columns 1, 12, 19, 33, 39, 66, and 80 of each line. These delimiters, in turn, form the columns for the mnemonic (listed exactly as it was previously within the first section), the scale factor (right-justified from column 17), the reference value (right-justified from column 31), the bit width (right-justified from column 37), and the units (left-justified from column 41). As with the previous two sections, blank "separator" lines may be employed in order to improve human-readability, and, for the same reason, it is also recommended to list the mnemonics in the same order in which they were declared within the first section, although this is by no means a requirement of the software. However, do note that any mnemonic whose corresponding data values are to be treated as character data must have its units listed as "CCITT IA5", which, again, is basically just a formal synonym for ASCII.