CHAPTER 2

GENERAL FORMAT DEFINITION

2.1. Format Structure. The format structure is constructed with information blocks. Information blocks provide control information and contain data. Figure 2-1 displays the general format of information blocks. A specific grouping of these blocks is used to create a product (see Section 2.1.2.1) and is considered a product data set.

2.1.1. Information Blocks Definition. An information block is a series of bytes identifying, controlling, or containing information used to create products. These blocks are characterized as control blocks, product definition blocks, data description blocks and data blocks.

2.1.1.1. Control Blocks. The control blocks are Product Identification, End of Product and parameter control blocks. The Product Identification Block is a standard block used for all products. Its purpose is to convey the information needed to uniquely identify each product so that appropriate processing routines may be initiated by the receiving system. The End of Product block is a standard block that signifies the end of the product data set. Other control blocks (see Chapter 4) provide product data set wide control of parameters in the data blocks.

2.1.1.2. Product Definition Block. The Product Definition block shall contain all information required to define the nature of the product being transferred (product area, scale, orientation, etc.).

2.1.1.3. Data Description Block. The Data Description block shall contain all information required to describe the contents of the data block(s) that follow (number and type of elements, element arrangement, units, etc.). The data description block shall be used when additional information about the structure and content of the data block(s) is required.

2.1.1.4. Data Block. The Data Block(s) shall contain the data in the format, units, etc., specified by the data description block, if not inherent in the data block itself.

2.1.2. Blocking Conventions. Multiple information blocks are used to create a product.

2.1.2.1. Product Data Set Structure. The product data set components shall be: a Product Identification block; a Product Definition block; one or more sets of control, data description, and data blocks; and each product data set shall be

terminated by an End of Product block. Figure 2-2 displays the general Product Data Set Structure.

2.1.2.2. Block Sequencing. The Product Identification block shall always be the first block in the product data set. The Product Definition Block, when required (see Section 3.4.1) shall immediately follow the Product Identification Block. Define Plot Parameters and Define Data Width/Field Width blocks may be interspersed with the data blocks and may appear anywhere in the product data set after the Product Identification block but before the data to which it applies. Multiple sets of Data Description blocks (when used), followed by one or more Data Blocks, may be used as required by the product originator to define all components of the product.

2.2. Block Format. Each block may contain the following fields: a LENGTH field, the MODE and SUBMODE fields, the DATA field, and a CHECKSUM field. These fields are defined in the notes following Figure 2-1. The LENGTH and CHECKSUM fields provide internal block information. The MODE and SUBMODE fields indicate the general content of the DATA field in the block.

2.2.1. Block Termination. The LENGTH field, if used, identifies the end of the block by providing a count of all byte pairs contained in the block. If the LENGTH field is not used, termination of a block can be accomplished by setting the most significant bit of the last byte in the data field. The originator must be able to guarantee that all other bytes in the data field have a zero in the most significant bit; otherwise, the LENGTH field will be necessary. In the case of text data, ETX plus ETB, or NULL, shall be used to end a block. (See Section 2.2.3.2.B.)

2.2.2. Block Size. Block length shall be variable but shall not exceed 4096 bytes, including the LENGTH and CHECKSUM fields when used. Multiple data blocks shall be used as required by the product originator to conform to the block length restriction and enhance circuit efficiency.

2.2.3. Block Format Conventions. The following general format conventions shall be observed within all blocks:

A. Bit numbering shall be right to left, starting with zero (the least significant bit).

B. Byte numbering shall be left to right, starting with byte zero.

C. All fields in the blocks shall be in one or more bytes unless otherwise specified in the block format or the data description block.

D. All block formats shall be arranged into two byte pairs with the bytes ordered left to right.

E. The LENGTH and CHECKSUM fields shall be used as required by the product originator. The presence or absence of the LENGTH and CHECKSUM fields shall be indicated by the flag in the LENGTH field. If the LENGTH and CHECKSUM fields are not used, the two left-most bits in byte zero of the block (i.e., the two most significant bits of the MODE byte) become the flag bits.

2.2.3.1. Graphic Display Information. The data contained in blocks which are display-oriented shall observe the following conventions:

A. All negative binary data shall be in two's complement form.

B. Binary data shall be right justified and zero filled.

C. All data is represented in octal notation unless otherwise noted.

2.2.3.2. Text Information. The data contained in text blocks (e.g. displayable messages) shall observe the following conventions:

A. All text data shall be seven (7) bit ASCII (ANSI X3.4-1977). Parity, if used, is transparent to these formats, except when the uppermost bit of the last data byte is being used to identify the last byte for recognition of block termination. (See Section 2.2.1.)

B. Text data shall be left justified within a field and blank filled. When using the text code form, one of two options will be used to terminate the text strings. The first option is to use the null (0) byte as the only termination for all text strings. In this case the characters ETX, ETB, and RS will not be used. The second option will use the RS character to separate records within a text data block. The control character ETB will be used at the end of blocks which are not the final block of a product data set. The character ETX will be the final character of an ASCII block which is the final block in a product data set. The most significant bit of the final byte (ETB or ETX) in the block may also be used for termination of a block when the previous bytes contain zeros in the most significant bit.

2.3. Coordinate System Conventions. Many of the products transmitted in the formats specified in this document use one of three coordinate systems to register and locate features at their proper position on a map or background. Once the coordinate system is selected, that system is then used for all subsequent coordinate references. The user needs to have a thorough understanding of the particular coordinate system being used and the implication that system has on proper data interpretation. The coordinate system in use for a product is indicated by a COORDINATE FLAG code found in the Product Definition Block. The three coordinate systems which can be used are:

Latitude/Longitude
Cartesian
Pixel.

The choice of coordinate systems affects interpretation of the following product registration information (applicable mode/submodes appear in parenthesis):

Reference M,N Coordinates	(4/20, 7/20)
M,N Maximum	(4/30)
M,N Center	(4/30)
M,N Coordinates	(4/1, 4/2, 4/3, 4/4, 4/5, 4/6, 4/7, 4/10, 4/11, 4/12, 5/1, 5/2, 5/3)
Delta M,N	(4/2, 4/5, 5/1)
ISTART, JSTART	(7/20)
IPOLE, JPOLE	(7/20)

The explicit meaning of the coordinate flag remains in effect until changed by a new coordinate flag presented in a new Product Definition Block within the product data set in question.

2.3.1. Latitude/Longitude Coordinate System (Coordinate Flag = 0). With this system, all product registration data use latitude and longitude to position features. All grid coordinate references appear in standard latitude and longitude on earth.

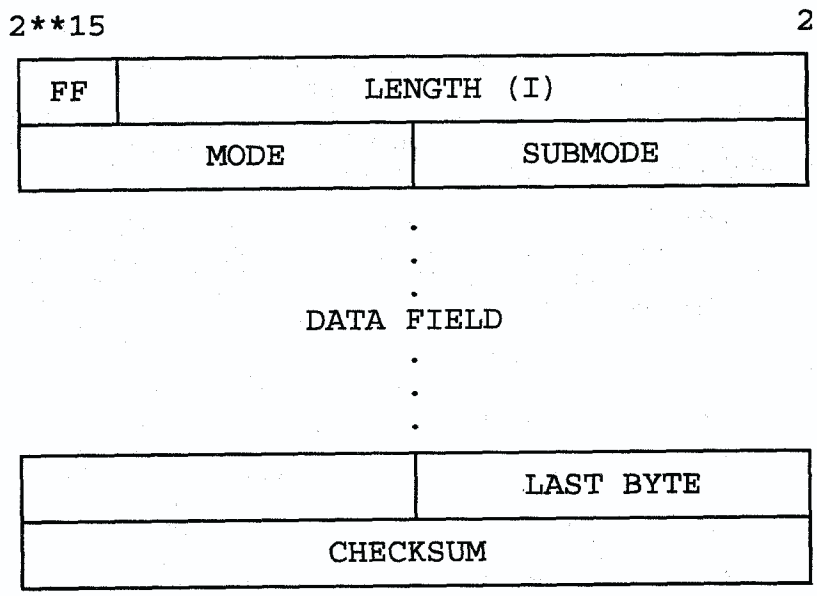
2.3.2. Cartesian Coordinate System (Coordinate Flag = 1). Many of the numerical models used in meteorology use some form of Cartesian coordinate grid system for mathematical manipulations. Because of this, it is convenient to output products with data registered to these Cartesian grid points. The array of points themselves are created projecting the earth onto a flat plane, selecting a coordinate system origin, and defining an array of

uniformly spaced points. The agencies involved in producing the products described in this document each have their own unique Cartesian coordinate grid systems but they are related. Mathematical equations can be used to convert from one system to the other but the user must have basic knowledge concerning the grid system of the source. This information includes:

Projection
Grid density
Indexing conventions
Origin point
Location of pole (in Cartesian coordinates)
Standard longitude
U,V-Wind component conventions.

Knowing these, the user can properly interpret the coordinate information imbedded in the product.

2.3.3. Pixel Coordinate System (Coordinate Flag = 2). In some cases, it is convenient to send product coordinate information relative to a logical display reference system, called pixel in this case. The origin (0,0) of the logical display space is the lower left corner and the indexing convention follows the right hand rule. Products are described relative to this logical display space in terms of horizontal (I-direction or X-direction) and vertical (J-direction or Y-direction) displacement from the origin. This pixel grid coordinate system can be translated by the user into whatever system needed for product manipulation and display.



NOTES:

1. FF . Flag: The Flag field is a two-bit LENGTH/CHECKSUM indicator. Possible combinations of these two bits are:

Flag Bits	*LENGTH field*	*CHECKSUM field*
00	*YES*	*YES*
01	*YES*	*NO*
11	*NO*	*NO*

If the most significant Flag bit is one (1), the least significant six bits of the left-hand byte will contain the Mode. The Flag bit combination 10 is not used as an indicator since there will never be a CHECKSUM field if the LENGTH field is not present. Blocks that contain non-text data shall always have a LENGTH field.

2. LENGTH: The LENGTH is the total number of two byte pairs in the current block, including the bytes containing the LENGTH and CHECKSUM fields if present.

3. The notation (I) indicates an unsigned integer quantity, e.g., LENGTH (I) indicates the LENGTH is an integer number.

4. MODE: The MODE indicator byte is contained in all transmission blocks and tells to the receiver the type of block being transferred. Note that the most significant bit will be a one (1) if LENGTH and CHECKSUM fields are not present.

5. SUBMODE: The SUBMODE is used to group each MODE into its logical subdivisions for the purpose of clarity in defining a block of data within a transmission mode.

Figure 2-1. General Block Format; Mode X, Submode Y

NOTES: Figure 2-1 (Cont.)

6. DATA FIELD: The DATA FIELD bytes contain information about the data and/or the data itself. Each block (i.e., MODE and SUBMODE combination) has information defined in this field for specific applications. The DATA FIELD will end on a two byte boundary, with binary data being zero filled and alphanumeric data being blank filled if necessary.

7. CHECKSUM: The CHECKSUM is a 16 bit field containing the two's complement of the arithmetic sum of all 16 bit byte pairs in the block with no end around carry. Adding all the byte pairs in a Mode/Submode that contains a CHECKSUM field will produce a sum equal to zero.

Decimal	Hex	Octal
5	0005	000005
769	0301	001401
86	0056	000126
54	0036	000066
-914	FC6E	176156

00	5
MODE	SUBMODE
YEAR	
SERIAL DAY	
CHECKSUM	

CHECKSUM = -914_{10} = $FC6E_{16}$ = 176156_{8}

Figure 2-1. (Cont.) General Block Format; Mode X, Submode Y

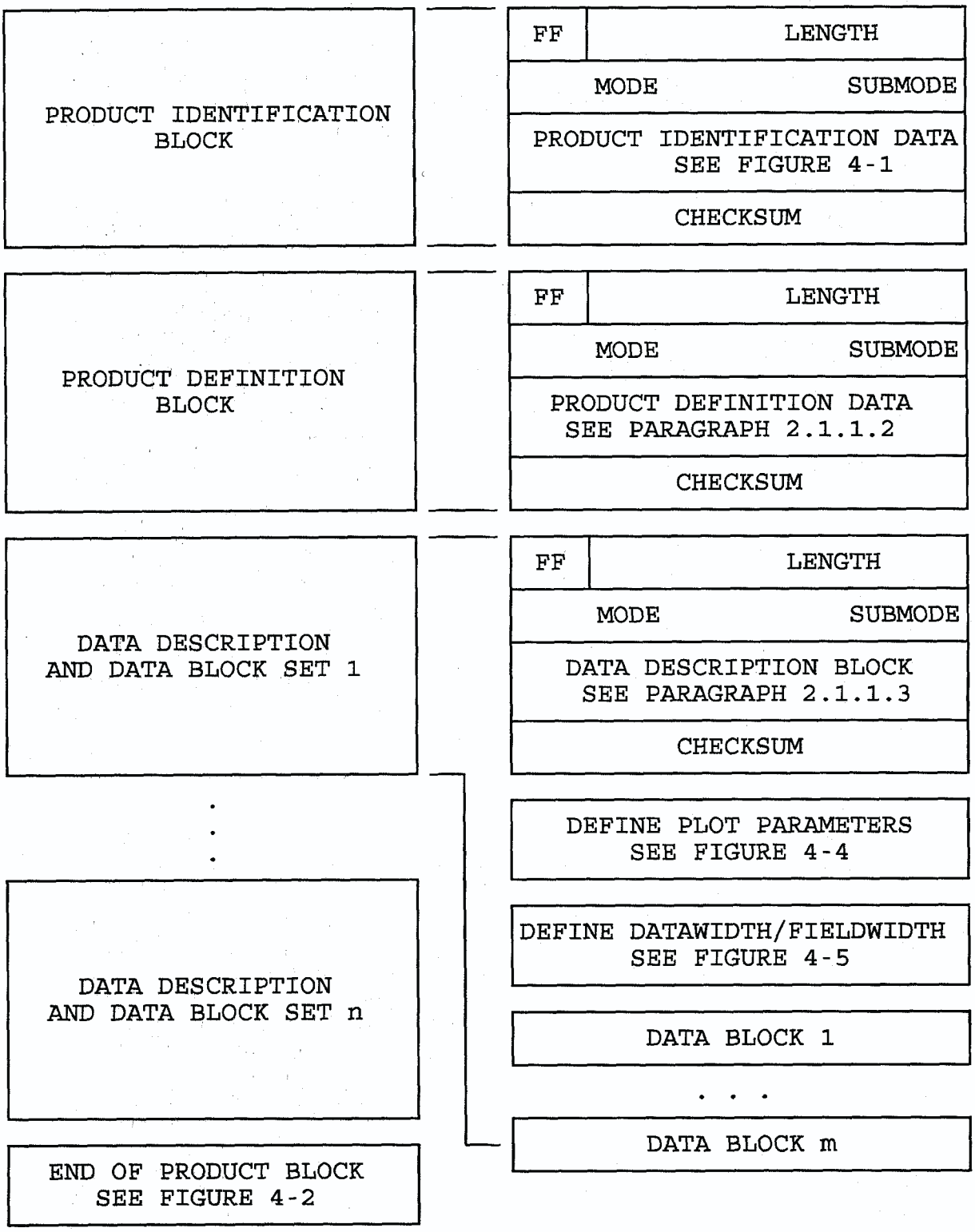


Figure 2-2. General Product Data Set Structure