METRIC

MIL-STD-2411
6 OCTOBER 1994

MILITARY STANDARD

RASTER PRODUCT FORMAT

AMSC N/A AREA MCGT

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited.

DEPARTMENT OF DEFENSE

1. This Military Standard is approved for use by all Departments and Agencies of the Department of Defense.

2. Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to: Director, Defense Mapping Agency, ATTN: TI, ST A-10, 8613 Lee Highway, Fairfax, VA 22031-2137 by using the Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document or by letter.

CONTENTS

1. SCOPE 1
1.1 Scope 1
1.2 Purpose 1
1.3 Application 1
1.4 Security 1

2. APPLICABLE DOCUMENTS 2
2.1 Government documents 2
2.1.1 Specifications, standards, and handbooks 2
2.1.2 Other Government documents, drawings, and publications 2
2.2 Non-Government publications 3
2.3 Order of precedence 4

3. DEFINITIONS 5

4. GENERAL REQUIREMENTS 6
4.1 Accuracy 6
4.2 Datum 6
4.3 Notation 6
4.3.1 Numbers 6
4.3.2 File structure 7
4.3.3 Other notation 10
4.4 Logical recording formats 10
4.4.1 Bit and byte order 10
4.4.1.1 Little endian 10
4.4.1.2 Big endian 10
4.4.2 Data types 11
4.4.2.1 Signed integer 11
4.4.2.2 Unsigned Integer 11
4.4.2.3 Real Value 12
4.4.2.4 ASCII Character String 14
4.4.2.5 Bit String 14
4.4.2.6 Byte String 14
4.4.2.7 Boolean 14
4.5 Recording standards 14
4.5.1.1 Compact Disk/Read-Only Memory (CD-ROM) 14
4.5.1.2 Erasable optical disk 15
4.5.1.3 8mm Magnetic tape cartridge 15
4.5.1.4 4mm Digital audio tape 15
4.5.1.5 Compact Disk/Recordable (CD-R) 15
4.5.2 Volume and file structures 15
4.5.3 Directory structures 15
4.5.4 File and directory naming convention 16
4.5.4.1 Directories 16
4.5.4.2 Table of contents file 16
4.5.4.3 External color/grayscale files 16
4.5.4.4 Frame Files 17

5.	DETAILED REQUIREMENTS	19
5.1	RPF concepts.	19
5.1.1	Compressed and uncompressed data.	19
5.1.2	Frames and subframes.	19
5.1.3	Frame and subframe sequence numbering, referencing, and recording.	19
5.1.4	Transparent pixels.	21
5.1.5	Frame and subframe masks.	21
5.1.6	Updating and replacing frames.	23
5.1.7	Reference designators.	26
5.1.8	Backward compatibility.	26
5.1.9	Packaging standards.	27
5.1.10	Relationship to the registered data standard.	27
5.1.11	Spectral bands.	28
5.1.12	Explicit areal coverage for attributes.	29
5.2	Data structure standards.	30
5.2.1	Table of contents file.	30
5.2.2	Frame file.	42
5.2.3	External color/grayscale file.	65
6.	NOTES	74
6.1	Intended use.	74
6.2	Acquisition requirements.	74
6.3	International standardization agreements.	74
6.3.1	International Standardization Agreements (STANAGs)	74
6.3.2	Quadripartite Standardization Agreements (QSTAGs)	74
6.3.3	Air Standardization Coordinating Committee Agreements (ASCC)	74
6.3.4	International MC&G Agreements.	74
6.3.5	Executive Orders.	74
6.3.6	InterAgency Agreements.	74
6.3.7	Other Documentation.	74
6.4	Subject term (key word) listing.	75
APPENDIX	BIT STRING, FIELD, AND GROUP NAME CROSS-REFERENCE	76
10	GENERAL	76
10.1	Scope	76
20	APPLICABLE DOCUMENTS	76
30	BIT STRINGS	76
40	FIELDS	76
50	GROUPS	82
INDEX		87

1. SCOPE

1.1 Scope. The Raster Product Format (RPF) is a standard data structure for geospatial databases composed of rectangular arrays of pixel values (e.g. in digitized maps or images) in compressed or uncompressed form. RPF is intended to enable application software to use the data in RPF format on computer-readable interchange media directly without further manipulations or transformation.

Each product category that represents a single instantiation of RPF, or a family of instantiations of RPF, shall be described in a separate product specification that makes appropriate reference to this RPF standard and its companion standard, MIL-STD-2411-1, which defines registered data values to be used with RPF files.

1.2 Purpose. The RPF is intended to define a common format for interchange of raster data between producers of such data in DoD and users of the data, to help facilitate interoperability among mission-critical systems.

1.3 Application. The Military Departments, Office of the Secretary of Defense, Organizations of the Joint Chiefs of Staff, and the Defense Agencies of the Department of Defense (collectively known as DoD components) shall use the information in this standard in preparing and accessing digital geographic data required or specified to be in RPF.

1.4 Security. This standard is UNCLASSIFIED. The procedures and processes presented herein may be used for classified processing where appropriate security provisions are added.

2. APPLICABLE DOCUMENTS

2.1 Government documents.

2.1.1 Specifications, standards, and handbooks. The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the current Department of Defense Index of Specifications and Standards (DODISS) and the supplement thereto, cited in the solicitation (see 6.2).

MIL-A-89007	Military Specification: ARC Digitized Raster Graphics (ADRG)
MIL-A-89027	Prototype Specification for ARC Digital Raster Imagery (ADRI)
MIL-STD-2407	Military Standard, Vector Product Format (VPF)
MIL-STD-2400	Military Standard, Text Product Standard (TPS)
MIL-STD-2411-1	Military Standard, Registered Data Values for Raster Product Format
MIL-STD-2411-2	Military Standard, Integration of Raster Product Format Files Into the National Imagery Transmission Format

(Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Standardization Documents Order Desk, Bldg. 4D, 700 Robbins Avenue, Philadelphia, PA 19111-5094).

2.1.2 Other Government documents, drawings, and publications. The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those listed in the current Department of Defense Index of Specifications and Standards (DODISS) and the supplement thereto, cited in the solicitation (see 6.2)

DMA TR 8350.2	Defense Mapping Agency World Geodetic System 84, 2d Edition

DMA TM 8358.1	Defense Mapping Agency Datums, Ellipsoids, Grids, and Grid Reference Systems

(Unless otherwise indicated, copies of federal and military specifications, standards, and handbooks are available from the Defense Mapping Agency, ATTN: PR, ST A-13, 8613 Lee Highway, Fairfax, VA 22031-2137).

2.2 Non-Government publications. The following document(s) form(s) a part of this document to the extent specified herein. Unless otherwise specified, the issues of the documents which are DoD adopted are those listed in the issue of the DODISS cited in the solicitation. Unless otherwise specified, the issues of documents not listed in the DODISS are the issues of the documents cited in the solicitation (see 6.2).

ANSI 3.41-1974	American National Standards Institute Code extension techniques for use with the 7-bit coded character set of American National Standard Code for Information Interchange
ANSI/IEEE Std. 754-1985	Institute of Electrical and Electronic Engineers IEEE Standard for Binary Floating Point Arithmetic
IEEE 1003.1	Institute of Electrical and Electronic Engineers Portable Operating System Interface for Computer Environments
ISO/IEC 10149	International Standards Organization Information technology -- Data interchange on read-only 120 mm optical data disks (CD-ROM)
ISO/IEC 10777	International Standards Organization Information technology 4 mm wide magnetic tape 2.2 cartridge for information interchange

ISO/IEC DIS 10089	International Standards Organization 130mm read-writeable optical media cartridge
ISO/IEC DIS 11319	International Standards Organization Information technology 8 mm wide magnetic tape cartridge for information interchange Helical scan recording
ISO/IEC 13346	Volume and File Structure of Write-Once and Rewriteable Media Using Non-Sequential Recording for Information Interchange
TBD	CD-Recordable Standard

(Application for copies of ANSI and ISO documents should be addressed to the American National Standards Institute (ANSI) Inc., 1430 Broadway, New YORK, NY 10018.)

(Non-Government standards and other publications are normally available from the organizations that prepare or distribute the documents. These documents also may be available in or through libraries or other informational services.)

2.3 Order of precedence. In the event of a conflict between the text of this document and the references cited herein (except for related associated detail specifications or specification sheets) the text of this document takes precedence. Nothing in this document, however, supersedes applicable laws and regulations unless a specific exemption has been obtained.

3. DEFINITIONS

This section is not applicable to this standard.

4. GENERAL REQUIREMENTS

a. The RPF shall be a general, adaptable format to encompass raster products in compressed or uncompressed form such as:

(1) non-polar and polar raster maps transformed from DMA's ARC Digitized Raster Graphics (ADRG) maps, as specified in MIL-A-89007; and

(2) imagery transformed from ARC Digital Raster Imagery (ADRI), as specified in MIL-A-89027.

b. RPF data recorded on random access interchange media shall be usable online at the user's discretion. The data structures must enable the user to process the data efficiently, whether it is used online or imported into the user's system, without further algorithmic transformation. The user may be permitted to relocate or reformat parts of the data to accommodate the needs of the receiving system. However, all such processes shall be completely reversible: the user shall be able to recover the original data on the interchange media by processing the relocated or reformatted data if necessary.

c. RPF shall accommodate new editions and updates to existing editions of data.

d. The RPF data structure shall enable receiving system software to be built with backward compatibility. That is, when receiving system software is constructed to read and process a given type of RPF data, it shall be capable of reading and processing that data type even if the RPF standard changes after the software is written, and if data of the given type is subsequently recorded in accordance with the changed standard.

e. RPF shall be integrated into the National Imagery Transmission Format (NITF) as defined in MIL-STD-2411-2.

4.1 Accuracy. The accuracy of individual RPF-compatible data products shall be specified in product specifications.

4.2 Datum. The horizontal datum for RPF data shall be WGS-84, as defined in DMA TM 8350.2.

The vertical datum for RPF data shall be as defined in individual product specifications.

4.3 Notation.

4.3.1 Numbers. All numbers defined herein shall be given in decimal notation, unless otherwise stated.

A number N given in hexadecimal notation shall be written as (N)H. A given hexadecimal digit may have any of the values (0, 1, ... 9, A, B, ...F).

4.3.2 File structure. The file structures given in section 5 employ the following notations:

<x> denotes an elementary field composed of bytes. Every byte field in this standard will be defined in terms of its data type, length, and domain.

/x/ denotes an elementary field or subfield composed of bits. Every bit field or subfield in this standard will be defined in terms of its data type, length, and domain (range of values).

[x] denotes a directory, file, or record -- i.e. a *group* of logical elements -- that is composed of ordered collections of fields and other groups of logical elements.

{x} denotes the start of a "level" of one or more elements in an ordered sequence. Each sequence of elements composing a file shall have an assigned numeric level {x}, where x = 1, 2, 3, 4, ... When a given group in a sequence of elements at level {x} is composed of subordinate elements, the sequence of subordinate elements shall be assigned to level {x+1}. The structure of each *file* and *record* is shown as an *ordered* indented list of its component parts, beginning with a level number. The structure of each *directory* is shown as an *unordered* indented list of its component parts, beginning with a level number. Each level of *indentation* corresponds to a level of *subordination*. For example, for a directory,

```
[directory 1]
     {1} (unordered)
     [file 1]
     [file 2]
     [directory 2]
```

indicates that [directory 1] contains [file 1], [file 2], and [directory 2], in any order.

Each file is defined completely in terms of its records and fields, recorded in the exact order indicated. For example:

```
[file 1]
     {1}
     <field a>
     <field b>
     [record 1]
          {2}
          <field c>
          <field d>
     {1}
     [record 2]
          {2}
          <field e>
          <field f>
          <field g>
```

indicates that [file 1] is composed of <field a>, followed by <field b>, followed by [record 1], followed by [record 2].

In turn, [record 1] consists of <field c> followed by <field d>; [record 2] is likewise composed of <field e>, <field f>, and <field g>, in that order.

In the figures that define the RPF file structures, the complete entry for a field will include its name, data type, and length in the format:

```
<field name>,data type:length or
/field name/,data type:length
```

The "data type" shall be a four-character abbreviation, as specified in TABLE 1 and 4.4.2 below. The "length" of a fixed-length byte field shall be an integer (number of bytes); the "length" of a fixed-length bit string shall be an integer (number of bits); the "length" of a variable-length field shall be designated as "var". For example,

```
{5}
[record 3]
     {6}
     <field h length>,uint:4
     <field h>,asci:var
     <field j>,asci:7
     /field k/,bits:12 (2)
```

indicates that [record 3] has the components <field h length>, <field h>, <field j>, and /field k/. Of these, <field h length> is a 4-byte unsigned integer (which also defines the length in bytes of <field h>); <field h> is a variable-length ASCII character string; <field j> is a 7-byte ASCII character string; and /field k/ is a bit string, 12 bits long, which occurs twice. Any logical group [x] or field <x> or field /x/ may be repeated. In general, the directories and files contain repeating groups of elements. Directories may contain repeating groups of files and

TABLE 1. Data Types and Their Abbreviations.

Data Type	Abbreviation (used in defining data structures)
ASCII string	asci
Bit string	bits
Byte string	byte
Boolean	bool
Integer (signed)	ints
Integer (unsigned)	uint
Real	real

other directories. Files may contain repeating groups of records and fields.

When an element occurs exactly once, it is shown on a line by itself without further elaboration. However, when an element is to be repeated, the range of repetition is shown in parentheses following the name of the element. For example,

```
[record 1] (1, ... 5)
[record 2]
```

indicates that [record 1] may occur 1, 2, 3, 4, or 5 times, but [record 2] occurs exactly once. The repetition of a group implies a repetition of each of its component parts, in turn. Note that components of any group may have their own repetition ranges as well. For example,

```
{4}
[record 1] (1, ... many)
    {5}
    <field a>,uint:2
    <field b>,asci:8 (1, ... 3)
    <field c>,byte:1 (32)
```

indicates that [record 1] occurs at least once, but can repeat an indeterminate number of times. *Each occurrence* of [record 1] will entail one occurrence of <field a>; 1, 2, or 3 occurrences of <field b> (each 8 bytes long); and 32 occurrences of <field c>, which is a byte string one byte long.

If a logical element may be present in some instances and not in others, then the range of repetition begins at 0. For example,

```
<field a>,real:4 (0, ... many)
<field b>,real:4 (0, 1)
```

indicates that <field a>, a 4-byte floating-point number, may be omitted entirely, or it may occur 1 or more times. <field b> may be omitted as well, but it will not occur more than once. This notation is extended to logical groups as well as to fields.

A directory may be composed of (contain) files and other directories. A file may be composed of records and fields. A record may be composed of fields and other records. Every occurrence of an [x] constitutes one occurrence of each of its component parts.

4.3.3 Other notation. The symbol "::=" means "is defined as." For example, the phrase

<x> ::= a 4-byte ASCII character string . . .

is equivalent to saying that <x> is defined as a 4-byte ASCII character string.

4.4 Logical recording formats.

4.4.1 Bit and byte order.

a. The default method of recording *numeric* (i.e. integer and real) data on interchange media shall adhere to the "big endian" convention as specified in 4.4.1.2 below. However, a given producer and a set of users may agree to exchange data in "little endian" fashion as specified in 4.4.1.1 below to optimize performance of a particular receiving system. The default byte ordering for numeric data fields in a given product shall be documented in its product specification.

b. Non-numeric data (e.g. pixel arrays or ASCII character strings) shall be recorded in the order in which the data is generated; the leftmost character shall always be recorded and read first.

c. The most significant bit in each byte of every field, regardless of data type, shall be recorded and read first, and successive bits shall be recorded and read in order of decreasing significance.

4.4.1.1 Little endian. In little endian format, the least significant byte in each numeric field shall be recorded and read first, and successive bytes shall be recorded and read in order of increasing significance. That is, if an n-byte field F is stored in memory beginning at address A, then the least significant byte of F shall be stored at A, the next at A+1, and so on. The most significant byte shall be stored at address A+n-1.

4.4.1.2 Big endian. In big endian format, the most significant byte in each numeric field shall be recorded and read first, and successive bytes shall be recorded and read in order of decreasing significance. That is, if an n-byte field F is stored in memory beginning at address A, then the most significant byte of F shall be stored at A, the next at A + 1, and so on. The least significant byte shall be stored at address A + n - 1.

4.4.2 Data types. Each field defined in section 5 below shall have one of the data types listed in TABLE 1 and specified in 4.4.2.1 through 4.4.2.7 below.

4.4.2.1 Signed integer. An N-byte signed integer field, containing B = (8 * N) bits, shall represent positive numbers in the range

$$0 \le x \le [2 \wedge (B - 1)] - 1$$

in the usual binary fashion and negative numbers in the range

$$(-1) * [2 \wedge (B - 1) - 1] \le x \le -1.$$

in twos complement form. That is, $-|x|$ shall be represented by $(2 \wedge B) - |x|$ in binary notation.

The number that is the twos complement of [2 ^ (B - 1)] shall be reserved as a null value.

Note: ^ denotes exponentiation.

For example, in a 2-byte signed integer, positive numbers are represented in the range $0 \le x \le 32,767$, and negative numbers in the range $-32,767 \le x \le -1$. The hexadecimal number (8000)H, the twos complement of -32,768, will be reserved as a null value in a 2-byte signed integer.

Signed integer fields in this standard shall be defined as 1-byte, 2-byte, or 4-byte signed integers, with the following null values: 1-byte signed integer, (80)H; 2-byte signed integer, (8000)H; and 4-byte signed integer, (8000 0000)H.

4.4.2.2 Unsigned Integer. An N-byte unsigned integer field, containing B = (8 * N) bits, shall represent positive numbers in the range

$$0 \le x \le (2 \wedge B) - 2$$

in the usual binary fashion.

The number (2 ^ B) - 1 shall be reserved as a null value, except as otherwise specified herein.

For example, in a 2-byte unsigned integer, positive numbers are represented in the range $0 \le x \le 65534$. The hexadecimal number (FFFF)H, corresponding to 65,535, will be reserved as a null value in a 2-byte unsigned integer.

4.4.2.3 Real Value. Real value fields shall be defined as 4-byte (single-precision) and 8-byte (double-precision) floating-point numbers, defined in accordance with ANSI/IEEE STD 754-1985. Using the data structure notation specified in 4.3.2 above, a <real value field> is defined as follows:

{1}
<real value field>
    {2}
    /sign bit/
    /biased exponent/
    /fraction/

(1) In a 4-byte real value, Bit 31 shall be the /sign bit/; Bits 30 through 23 shall constitute the /biased exponent/; and Bits 22 through 0 shall constitute the /fraction/. The number represented by a 4-byte <real value field> shall be

x = [(-1) ^ /sign bit/] * [2 ^ (/biased exponent/ - 127)] * [1 + /fraction/]

where:

/sign bit/ ::= 0 where <real value field> represents a positive number, and ::= 1 where <real value field> represents a negative number.

/biased exponent/ ::= an integer in the range 0 to 255; when the bias is subtracted, the actual exponent represented is in the range -127 to +128.

/fraction/ ::= a number in the range 0 to {1 - [2 ^ (-23)]}. The value of the most significant bit of /fraction/ is 2 ^ (-1) and the value of the least significant bit is 2 ^ (-23).

The values plus and minus zero shall be represented as follows:

/biased exponent/ ::= 0
/fraction/ ::= 0
/sign bit/ ::= 0 to represent a value of +0, and /sign bit/ ::= 1 to represent a value of -0.

The values plus and minus infinity shall be represented as follows:

/biased exponent/ ::= 255
/fraction/ ::= 0
/sign bit/ ::= 0 to represent plus infinity and /sign bit/ ::= 1 to represent minus infinity.

Null values in 4-byte floating-point data fields shall be coded to equal plus infinity, as defined above.

The value "not a number" (NaN), resulting from an invalid operation on a floating-point number (such as divide by zero), shall be represented as follows:

/biased exponent/ ::= 255
/fraction/ > 0
/sign bit/ ::= 0 or 1

(2) In an 8-byte real value, Bit 63 shall be the /sign bit/; Bits 62 through 52 shall constitute the /biased exponent/; and Bits 51 through 0 shall constitute the /fraction/. The number represented by an 8-byte real value shall be

x = [(-1) ^ /sign bit/] * [2 ^ (/biased exponent / - 1023)] * [1 + /fraction/]

where:

/sign bit/ ::= as defined in 4.4.2.3(1) above;

/biased exponent/ ::= an integer in the range 0 to 2047; when the bias is subtracted, the actual exponent represented is in the range -1023 to +1024.

/fraction/ ::= a number in the range 0 to {1 - [2 ^ (-52)]}. The value of the most significant bit of /fraction/ is 2 ^ (-1) and the value of the least significant bit is 2 ^ (-52).

The values plus and minus zero shall be represented as follows:

/biased exponent/ ::= 0
/fraction/ ::= 0
/sign bit/ ::= 0 to represent a value of +0, and /sign bit/ ::= 1 to represent a value of -0.

The values plus and minus infinity shall be represented as follows:

/biased exponent/ ::= 2047
/fraction/ ::= 0
/sign bit/ ::= 0 to represent plus infinity and /sign bit/ ::= 1 to represent minus infinity.

Null values in 8-byte floating-point data fields shall be coded to equal plus infinity, as defined above.

The value "not a number" (NaN), resulting from an invalid operation on a floating-point number (such as divide by zero), shall be represented as follows:

/biased exponent/ ::= 2047
/fraction/ > 0
/sign bit/ ::= 0 or 1

4.4.2.4 ASCII Character String. An ASCII character string field shall comprise one or more bytes, each coded in accordance with ANSI 3.41-1974 , section 6. Each ASCII character string shall be left justified and filled with trailing SPACE characters unless otherwise specified. An ASCII character string of all spaces will indicate a null value.

4.4.2.5 Bit String. A bit string field shall comprise one or more bits -- not necessarily a multiple of 8 bits, which would constitute a byte -- in which each bit, named in accordance with 4.3.1.e above, shall be coded with the values 0 and 1 and interpreted independently of every other bit in the field. Alternatively, a bit string may be composed of groups of bits, with each group coded and interpreted independently of every other group in the string.

4.4.2.6 Byte String. A byte string field shall consist of one or more bytes whose encoding is determined by a given product specification but not specified in this standard. For example, a given product specification may interpret a field -- that is specified as a byte string field herein -- as an integer, an ASCII string, a real value, or as a unique type, i.e. one that is not defined herein, depending on the nature of the product.

4.4.2.7 Boolean. A Boolean field shall be a one-byte field coded as follows: (00) hexadecimal ::= false; (FF) hexadecimal ::= true. No other bit values shall be coded.

4.5 Recording standards. To guarantee interoperability between originators and receivers of RPF data, it is intended that the data will be recorded on high-density random access and sequential media defined by international standards.

RPF data may also be recorded on other media for archiving, for internal use in a system, or for interchange in accordance with agreements that may be reached between a given originator and a given group of receivers; however, in such cases interoperability shall not be guaranteed for third-party receivers who are not parties to such agreements.

4.5.1 Media. RPF data shall be recorded as specified herein on any of the media types described below. Other standard media may be added in the future to reflect progress in recording media technology.

4.5.1.1 Compact Disk/Read-Only Memory (CD-ROM). The data shall be recorded on read-only 120mm optical data disks as defined in ISO/IEC 10149.

4.5.1.2 Erasable optical disk. The data shall be recorded using a continuous composite servo tracking method on 130 mm (5.25-in.) erasable optical disks (EODs), as defined in ISO/IEC DIS 10089.

The User Zone on each EOD shall be formatted as defined in ISO/IEC DIS 10089, Sections 16 and 17, with 512 bytes per sector.

4.5.1.3 8mm Magnetic tape cartridge. The data shall be recorded on 8mm wide magnetic tape cartridges manufactured in accordance with ISO/IEC DIS 11319.

4.5.1.4 4mm Digital audio tape. The data shall be recorded on 4mm wide magnetic tape cartridges manufactured in accordance with ISO/IEC DIS 10777.

4.5.1.5 Compact Disk/Recordable (CD-R). The data shall be recorded on 120mm optical data disks as defined in TBD.

4.5.2 Volume and file structures.

a. The volume and file structure for data recorded on CD-ROM and CD-R shall be as defined in ISO 9660.

Note: The nature of the ISO 9660 CD-ROM standard is such that platforms will present the files and directory names differently. As an example, although the files are written to the CD-ROM in upper case (as defined in the standard), some file systems will present the data in lower case letters. In addition, the following differences in the presentation of the file and directory names may also appear on various platforms:

- Filenames with extensions may be appended with a semicolon-1 (;1).

- Filenames without extrensions may be appended by a single period (.) or with a period-semicolon-1(.;1).

The table of contents file for RPF products includes the pathnames to the frame files. The pathnames and frame files in the table of contents file are written in upper case, and the directory delimiters are given as forward slashes (/). Developers of software for CADRG will be required to understand the format of the pathnames in the table of contents file, as defined in this section, in order to properly use the table of contents file.

b. The preferred volume and file structure for data recorded on read-write and write-once/read-many random access media (i.e. disks) shall be the Non-Sequential Recording Format, as defined in ISO 13346.

c. User data on sequential media (i.e. tapes) shall be formatted in accordance with the extended tar format, as defined in IEEE 1003.1, paragraph 10.1.1.

4.5.3 Directory structures. The overall format and structure of directories shall be in accordance with Section 5 of IEEE Standard 1003.1.

A given random access volume shall contain the following directory structure:

```
[rpf root directory] (unordered)
     {1} (unordered)
     [table of contents file]
     [lookup table directory] (0, 1)
          {2} (unordered)
          [external color/grayscale file] (1, ... many)
     {1} (unordered)
     [frame directory] (0, ... many)
          {2}  (unordered)
          [frame file] (0, .. many)
          [subordinate directory] (0, ... many) (unordered)
```

4.5.4 File and directory naming convention. The unqualified name of each RPF file shall be in the industry standard "8.3" format: up to 8 alphanumeric characters, followed by a period, followed by an extension of up to three alphanumeric characters. The unqualified name of each RPF directory shall be up to 8 alphanumeric characters.

4.5.4.1 Directories. Directories on RPF interchange volumes are intended to help locate the data on a given interchange volume. At the discretion of the implementor, receivers of the data may relocate the directory tree found on the interchange media anywhere within their own file system hierarchies for processing.

a. The [rpf root directory] on a given volume shall be named "RPF". Every RPF volume shall have an [rpf root directory], plus at least one subordinate [frame directory] or [lookup table directory].

b. All [external color/grayscale file]s shall be stored on the interchange media in the "RPF/LOOKUP" directory.

c. [frame file]s shall be stored in a separate directory hierarchy under the RPF directory. The producer shall determine a strategy for choosing the hierarchical structure, the name of each directory at each level in the hierarchy in a given volume, and a method for assigning [frame file]s to specific directories. Each [subordinate frame directory] shall have the same structure as the [frame directory]. Each [frame directory] shall contain at least one [frame file], in the [frame directory] itself or in a [subordinate frame directory].

4.5.4.2 Table of contents file. The [table of contents file] shall be named "A.TOC"; its pathname on the interchange media shall be "RPF/A.TOC".

4.5.4.3 External color/grayscale files.

a. All [external color/grayscale file] file names shall be logically coded as follows:

```
[file name]
    {1}
    <reference designator>,asci:4
    <period>,asci:1
    <extension>,asci:3
```

b. The file name described in 4.5.4.3.d below shall also be recorded in the [header section] of the given [external color/grayscale file].

c. A central authority (i.e. DMA) shall assign each producer of RPF [frame file]s a block of <reference designator> "numbers" for its exclusive use. Each producer shall be responsible for generating a <reference designator> value that makes the [file name] unique for each [external color/grayscale file] that it produces.

d. [file name]s shall contain the following logical elements, listed in alphabetical order:

(1) <extension> ::= a 3-byte ASCII character field ::= "LUT".

(2) <period> ::= a 1-byte ASCII character field ::= "."

(3) <reference designator> ::= a 4-byte ASCII character field chosen to make a given file name unique. Each byte of <reference designator> shall have one of the following possible values: capital letter, A through Z, excluding I and O. The <reference designator> may be regarded as a radix 24 number that can be assigned any of 24^4 possible values.

4.5.4.4 Frame Files.

a. All RPF [frame file] names shall be logically coded as follows:

```
[file name]
        {1}
        <reference designator>,asci:8
        <period>,asci:1
```

<data series and zone>,asci:3

b. The file name described below shall also be recorded in the [header section] of the given [frame file].

c. A central authority (i.e. DMA) shall assign each producer of RPF [frame file]s a block of <reference designator> "numbers" for its exclusive use. Each producer shall be responsible for generating a <reference designator> value which, combined with a given <data series and zone> shall make the [file name] unique for each [frame file].

d. [file name]s shall contain the following logical elements, listed in alphabetical order:

(1) <data series and zone> ::= a 3-byte ASCII character field indicating the data type of the given file, encoded as specified in MIL-STD-2411-1, section 5.1.5.

(2) <period> ::= a 1-byte ASCII character field ::= "."

(3) <reference designator> ::= an 8-byte ASCII character field chosen to make a given file name unique for a given <data series and zone> value. Each new, replacement, and update [frame file] shall have a unique reference designator for a given value of <data series and zone>. Each byte of <reference designator> shall have one of the following possible values: any digit from 0 through 9; or any capital letter, A through Z, excluding I and O. The <reference designator> may be regarded as a radix 34 number that can be assigned any of 34^8 possible values.

5. DETAILED REQUIREMENTS

5.1 RPF concepts. The RPF is designed to accommodate digital map and image products in compressed or uncompressed form for interchange between producers of the data such as DMA and users of the data in the military services. The following concepts pertaining to the construction and use of RPF files are described below: accommodation for compressed and uncompressed data; use of frames, subframes, and masks; transparent pixels; updating and replacing files; backward compatibility; relationship of RPF to the Vector Product Format and the Text Product Standard; packaging standards; and the relationship of the RPF to its companion standard for registered data values.

5.1.1 Compressed and uncompressed data. The essence of an *uncompressed* RPF data file is a matrix of data element values such as color values in a digitized map or grayscale values in a grayscale image. For a *compressed* RPF data file, the matrix of element values can be decompressed by some well-defined algorithm to recover a corresponding matrix of color or grayscale values. The producer is responsible for compressing the data using an algorithm whose essential parameters are defined in the [frame file]; the user is responsible for decompressing the data and using it in a meaningful way. This RPF standard does not specify specific compression or decompression algorithms; it simply provides a data structure that will accommodate the element values and the accompanying information, such as lookup tables and parameter lists, that the user needs to perform the decompression. The product specification for a given RPF-compatible product type will contain the details of applicable compression and decompression algorithms.

5.1.2 Frames and subframes. In general, the matrix of data in a given [frame file] will be organized into a matrix of subframes, each comprising, in decompressed form, a fixed number of data elements (e.g. 256 x 256 pixels in a typical map [frame file]). A typical frame, in turn, will be composed of a matrix of such subframes (e.g. 6 x 6 subframes).

5.1.3 Frame and subframe sequence numbering, referencing, and recording.

a. Each frame shall provide information about a given geographic area defined in the [coverage section] of the [frame file]. In a series of frames of *identical size and scale in a given geographic zone*, each frame can be distinguished from the others within the series by simply naming one of its four corners. By convention, the southwest (lower left) corner will be used to characterize a frame in such a context. For example, the <geographic location> defined for a [frame file] in a [frame file index record] in the [table of contents file] (see 5.2.1 below) will be the location of the southwest (lower left) corner of the [frame file].

b. Each boundary rectangle defined in the [boundary rectangle section] of a [table of contents file] will act as a logical container for a rectangular "virtual matrix" of frames (see 5.2.1 below). Individual frames may be referenced by their (row, column) position within such a matrix.

(1) By convention, the southernmost (lowest) row of every frame matrix shall be Row 0, and row numbering shall increase in the northerly (up) direction. In a matrix with N rows, the northernmost (topmost) row shall be numbered Row N-1.

(2) Similarly, in a matrix with M columns, the westernmost (leftmost) column in such a matrix shall be Column 0, and the easternmost (rightmost) column shall be Column M-1.

c. The frame itself will act as a logical container for one or more subframes, arranged within a rectangular matrix (see 5.1.2 above and 5.2.2 below). Individual subframes may be referenced by their (row, column) position within such a matrix.

(1) By convention, the northernmost (top) row of every subframe matrix shall be Row 0, and row numbering shall increase in the southerly (down) direction. In a matrix with N rows, the southernmost (bottom) row shall be numbered Row N-1.

(2) Similarly, in a matrix with M columns, the westernmost (leftmost) column in such a matrix shall be Column 0, and the easternmost (rightmost) column shall be Column M-1.

(3) Within a [frame file], non-empty subframes shall be recorded in row major order, beginning with Row 0, Column 0. (Empty subframes shall not be recorded.)

d. Likewise, the subframe will act as a logical container for rows and columns of /image code/s representing pixels in compressed or uncompressed form, arranged within a rectangular matrix (see 5.2.2 below). Individual /image code/s may be referenced by their (row, column) position within such a matrix.

(1) By convention, the northernmost (top) row of every /image code/ matrix shall be Row 0, and row numbering shall increase in the southerly (down) direction. In a matrix with N rows, the southernmost (bottom) row shall be numbered Row N-1.

(2) Similarly, in a matrix with M columns, the westernmost (leftmost) column in such a matrix shall be Column 0, and the easternmost (rightmost) column shall be Column M-1.

(3) /image code/s shall be recorded in row major order, beginning with Row 0, Column 0.

5.1.4 Transparent pixels. In general, each image data point in a subframe will represent the color, intensity, or value of a variable that characterizes the given point. However, in some cases, data may be missing, or not available at a given geographic point in a subframe. In such cases, a so-called "transparent pixel" or "null" value shall be assigned to the data at the given point; users shall treat transparent pixels as null values, and not as numeric quantities. (See the related discussion of null values in 4.4.2.2 above.) A specific RPF-compatible product type that use compressed [frame file]s may contain a product-specific method of designating parts of the compressed data that represents transparent pixels, or a product-specific method of designating output (decompressed) transparent pixels, or both.

5.1.5 Frame and subframe masks.

a. As noted in 5.1.3 above, a frame is a logical container for one or more subframes. When a frame is composed of multiple subframes, some subframes may contain data and others may not, at a given time. The non-empty subframes shall be recorded on the interchange volume, and the empty subframes shall be omitted. Therefore, a mechanism is needed to designate which subframes are present and which (if any) are omitted from the interchange volume.

(1) As noted in 5.1.3 above, the non-empty subframes shall be recorded in a contiguous sequence in row major order. If, for example, a frame contains a matrix of 3 x 3 variable-length subframes, and the subframes for (Row 0, Column 0) and (Row 1, Column 2) are empty, then the lengths of the recorded and unrecorded subframes might be as shown in TABLE 2.

TABLE 2. Example of subframe lengths.

Subframe Row	Subframe Column	Subframe Length	Cumulative Length
0	0	(0)H	(0)H
0	1	(0000 1100)H	(0000 1100)H
0	2	(0000 1200)H	(0000 2300)H
1	0	(0000 1400)H	(0000 3700)H
1	1	(0000 0F00)H	(0000 4600)H
1	2	(0)H	(0000 4600)H
2	0	(0000 0C00)H	(0000 5200)H
2	1	(0000 0E00)H	(0000 6000)H
2	2	(0000 1100)H	(0000 7100)H

(2) When a [frame file] is recorded with one or more empty subframes, the [frame file] shall include a [subframe mask table] that relates the row and column number of each subframe to the address (byte number) of the /image code/s comprising each subframe.

(3) The mask will consist of an array of <subframe offset> entries, in row major order, one for every subframe in the frame, whether or not it is empty.

(4) If a subframe is not empty, its <subframe offset> entry will be the address of the first byte of the subframe, measured from the beginning of the [spatial data subsection] in the [frame file]; if the subframe is empty (and therefore not recorded), its <subframe offset> shall be assigned a null value. In the example cited above, the <subframe offset> values in the [subframe mask table] will be as shown in TABLE 3. In the given example, the entire length of the [spatial data subsection] (recorded in the [location section]) containing the seven recorded subframes will be (7100)H. Using the information recorded in the [location section], the user will be able to calculate the length of Subframe (2, 2) as (1100)H. Similarly, using the <subframe offset> data in TABLE 3, the user will be able to calculate the lengths of the other six recorded subframes. For example, the length of Subframe (1, 1) = (4600)H - (3700)H = (0F00)H.

TABLE 3. Example of [subframe mask table] entries.

Subframe Row	Subframe Column	<subframe offset> value
0	0	(FFFF FFFF)H
0	1	(0000 0000)H
0	2	(0000 1100)H
1	0	(0000 2300)H
1	1	(0000 3700)H
1	2	(FFFF FFFF)H
2	0	(0000 4600)H
2	1	(0000 5200)H
2	2	(0000 6000)H

(5) The receiver of a [frame file] will be able to use the [subframe mask table] to determine the (row, column) locations -- and therefore the geographic locations -- of the subframes that are present in the [frame file].

(6) If all of the subframes in a [frame file] are non-empty and equal in length, then the [subframe mask table] may or may not be recorded. (The product specifications for a given RPF-compatible product will establish whether or not the [subframe mask table] is always recorded.) The receiver in this case will be able to compute the relationship between the implicit sequence number of each subframe and its (row, column) position.

b. A non-empty subframe may contain one or more transparent pixels, as described in 5.1.4 above. In such cases, a [transparency mask table] shall be recorded in the [frame file] to indicate each non-empty subframe that contains at least one transparent pixel.

(1) The structure of the [transparency mask table] shall be identical to that of the [subframe mask table] described above. For example, in the example cited above, if the subframe in (Row 2, Column 1) and the subframe in (Row 2, Column 2) each contain one or more transparent pixels, and none of the other subframes contain any transparent pixels, then the entries in the [transparency mask table] will be as shown in TABLE 4.

TABLE 4. Example of [transparency mask table] entries.

Subframe Row	Subframe Column	<subframe offset> value
0	0	(FFFF FFFF)H
0	1	(FFFF FFFF)H
0	2	(FFFF FFFF)H
1	0	(FFFF FFFF)H
1	1	(FFFF FFFF)H
1	2	(FFFF FFFF)H
2	0	(FFFF FFFF)H
2	1	(0000 5200)H
2	2	(0000 6000)H

(2) If none of the subframes in the [frame file] contain at least one transparent pixel, then the [transparent pixel mask table]; may or may not be recorded, as defined in product specifications.

c. If all subframes are present and at least one subframe contains one or more transparent pixels, then the [transparency mask table] shall be recorded and the [subframe mask table] may or may not be recorded.

d. If one or more subframes are empty and at least one of the non-empty subframes contains one or more transparent pixels, then the [transparency mask table] and the [subframe mask table] shall be recorded.

e. If one or more subframes are empty and none of the subframes in the [frame file] contain at least one transparent pixel, then the [subframe mask table] shall be recorded and the [transparency mask table] may or may not be recorded.

f. If all subframes are present and none of the subframes in the [frame file] contain at least one transparent pixel, then the entire [mask subsection] may or may not be recorded.

5.1.6 Updating and replacing frames.

a. The RPF design accommodates update and replacement of existing raster products. To incorporate this capability, each [frame file] shall be characterized as one of three types:

(1) An original [frame file], representing the first edition of a given type of raster data at a given scale or resolution and covering a given geographic area;

(2) A [frame file] that is a new edition, completely replacing all previous editions of the [frame file] for the given data type, scale or resolution, and coverage;

(3) An update or patch that replaces one or more of the subframes comprising the parent edition of the [frame file], but leaves the other subframes unchanged.

b. Each [frame file] of a given data type shall be assigned a unique [file name], which can be referenced when an update or replacement occurs as described in 4.5.4.4 above. Moreover, each update or replacement [frame file] will include a small table that names all of the relevant [frame file]s -- original, replacement, or update -- that are the ancestors of the current [frame file] and identifies the relationships among them.

c. The information contained in successive updates to a given parent [frame file] shall be cumulative. That is, each update to a given parent shall contain new update information (e.g. inserting a new tower) and shall also contain all of the previous updates to that same parent (e.g. a previously reported new power line and a change in the label of an elevation contour) that were published since the parent was issued.

For example, suppose that DMA produces a raster [frame file] with 55:1 compression at 1:1M scale in 1993 for the geographic region 36 to 40 degrees north latitude and 112 to 116 degrees west longitude, with <reference designator> ABCD1234. In 1994, the same producer publishes a replacement [frame file] for that entire region, with <reference designator> EF56GH78. Then the [replace/update section] for frame EF56GH78 will have one record, indicating that the current [frame file] completely replaces frame number ABCD1234, as shown in TABLE 5.

TABLE 5. Example of [replace/update record]s for second edition of a [frame file].

New [frame file] designator	Old [frame file] designator	Replace/Update Status
EF56GH78	ABCD1234	3 (Replace)

Suppose further that in 1995 an update to frame EF56GH78 is published in frame J901K234. Then the [replace/update section] for frame J901K234 will have two records, as shown in TABLE 6: the first will indicate that the current frame updates frame EF56GH78; the second record will indicate that frame number EF56GH78, in turn, replaces frame number ABCD1234.

Later in 1995, the producer issues yet another update to frame EF56GH78, in frame L567890M, which contains a new change and *also duplicates* the change that was first specified in frame number J901K234. The [replace/update section] for frame L567890M has three records, as shown in TABLE 7: the first two indicating that the current [frame file] updates frame EF56GH78 and supersedes frame J901K234; the third indicating that frame number EF56GH78, in turn, replaces frame number ABCD1234.

TABLE 6. Example of [replace/update record]s for the first update to the second edition of a [frame file].

New [frame file] designator	Old [frame file] designator	Replace/Update Status
J901K234	EF56GH78	1 (Update)
EF56GH78	ABCD1234	3 (Replace)

TABLE 7. Example of [replace/update record]s for the second update to the second edition of a [frame file].

New [frame file] <reference designator>	Old [frame file] <reference designator>	Replace/Update Status
L567890M	EF56GH78	1 (Update)
L567890M	J901K234	2 (Supersede)
EF56GH78	ABCD1234	3 (Replace)

In other words, each [frame file], other than an original [frame file] with no ancestors, will contain the entire replacement and update history of the current [frame file], so that receivers who have *any* of the previous editions or updates will know how to handle the current [frame file]. (For example, holders of frame EF56GH78 will be able to update it with frame L567890M even though they may never have received frame J901K234; holders of frame ABCD1234 who don't have frame EF56GH78 will be able to order frame EF56GH78 from the producer.)

Updates will *not* extend across multiple editions of compressed [frame file]s; that is, it will not be possible for J901K234 and L567890M to update ABCD1234.

In addition, the update history for a given [frame file] will pertain only to the current edition. When a new edition of the complete [frame file] is released, the [replace/update record]s for all previous full editions of the file will be kept, but updates to previous editions will not be carried forward.

This situation is illustrated in TABLE 8. In 1996, the producer publishes a complete replacement for EF56GH78, in frame NPQR4321. The records show that NPQR4321 replaced EF56GH78, and EF56GH78 replaced ABCD1234.

All [replace/update record]s shall be stored in descending order by <replace/update status>, and then in descending order by new <reference designator>, and then in descending order by old <reference designator>.

5.1.7 Reference designators. The central producer (DMA) will issue itself and every other authorized producer a block of <reference designator>s for its use. Each producer will be responsible for guaranteeing that a unique <reference designator> is assigned to each [frame file] -- and each external color/grayscale file -- that it publishes.

TABLE 8. Example of [replace/update record]s for the third edition of a [frame file].

New [frame file] <reference designator>	Old [frame file] <reference designator>	Replace/Update Status
NPQR4321	EF56GH78	3 (Replace)
EF56GH78	ABCD1234	3 (Replace)

5.1.8 Backward compatibility. The RPF structure incorporates mechanisms that facilitate backward compatibility in the evolution of the structure itself. An underlying premise is that the structure will evolve through growth, rather than replacement. Therefore, backward compatibility is achieved as follows:

- By enabling new sections to be added to any file, subject to approval of the configuration control organization, without affecting the structures of existing files; and

- By enabling new subsections and new fields and tables to be added to existing sections without affecting the structure of the remainder of the section.

As a result, properly written software that can read one RPF file will be able to read any other RPF file, even though it may not know how to process some new sections or parts of sections.

For example, the [header section] of each RPF file (see FIGURE 2, FIGURE 3, or FIGURE 4) will contain a <location section location> field. RPF software should be written to always use the value in this field to determine where the [location section] begins. Then, if additional fields or groups are added to the end of the [header section] in a future implementation of this standard, pre-existing software will be able to find the [location section] and ignore the new fields. Moreover, RPF volumes written to the present standard will be still be usable. The intention is that any software that can read (i.e. parse) the [header section] of a given RPF file will always be able to read at least the beginning of the [header section] of any other RPF file, find the [location section], and therefore find every other section in the file.

Similarly, if a future version of this standard adds new fields to a given section of a file, pre-existing software employing the built-in offset fields will still be able to locate the already-defined tables and groups in the given section.

The various <... record length> fields referring to fixed-length records (such as <boundary rectangle record length> in the [table of contents file]) are likewise intended to facilitate backward compatibility, enabling any future version of this standard to add new fields to the referenced record (the [boundary rectangle record] in this example) without affecting the capability of then-existing software to read each record.

The completely general structure of the [location section] will enable new types of sections to be added to the file in the future with minimal impact on existing software. That is, the user's program will be required to identify each section in order to process it; if it encounters a section that it doesn't recognize, it will not be prevented, necessarily, from processing the remaining sections of the file that it does recognize. It will be possible, under configuration management, to add new fields or record types to satisfy future requirements without adversely affecting the capability of existing software to read and process data. The user's software should be required to ignore any record types that it does not recognize.

5.1.9 Packaging standards. A given RPF volume may contain a mix of raster product types in the RPF directory (see 4.5.3 above), encompassing a variety of scales (or resolutions) and compression techniques. Producer policies will determine strategies for allocating specific product types to specific volumes.

The RPF shall be treated as a companion standard to the Vector Product Format (VPF) defined in MIL-STD-2407 and the Text Product Standard (TPS) defined in MIL-STD-2400. It is intended that VPF-compatible data files and TPS-compatible data files may be recorded on the same media as RPF-compatible data files, in separate directories outside the RPF directory, at the discretion of the producer.

5.1.10 Relationship to the registered data standard. The codes to be used in a number of fields in RPF files, such as <component id> and <security classification>, will be defined in MIL-STD-2411-1, which is intended to be a companion to this standard. The codes that pertain to a given RPF-compatible data product will be identified in the specification for the given product. MIL-STD-2411-1 will be updated periodically to reflect additional code values. DMA will maintain MIL-STD-2411-1 to ensure the following: that the meanings of the codes are universally understood; that a given code for a given field shall have only one meaning, no matter which application or product the code value is used for; and that a mechanism exists for defining and disseminating new codes as the applications, products, and this standard evolve.

5.1.11 Spectral bands. In general, a raster product consists of a series of pixels or data elements that represent -- in compressed or uncompressed form -- a matrix of values of some variable of interest (e.g. the color of a point on a map). To accommodate large raster databases, the data are partitioned into frames, and then into subframes. In a simple raster product, each pixel or data element corresponds to a single-valued quantity, such as a grayscale value.

But in more complex products, such as color maps and images, the variable of interest may be expressed in terms of several components. For example, a typical RGB color image has distinct red, green, and blue components, each described by an 8-bit (1-byte) variable. One could record, in each subframe, the series of values as a sequence of 3-byte fields, each representing the R, G, and B components, in turn, of a single pixel. This is typically called interleaved, or band interleaved by pixel format.

As another alternative, one could store in a single table, within each subframe, all of the R components for a subframe line, followed by the B components for a subframe line, and then the G components for the subframe line. This pattern would be repeated for each subframe line. This is typically called band interleaved by line format.

Alternatively, within each subframe, one could segregate all of the R components into one table (called a spectral band table), the G components into another spectral band table *with the identical structure*, and the B components into a third spectral band table, also with the identical structure. This is defined as band interleaved by subframe format.

In each of these structural variants, the subframe is the highest organizational unit. That is, all the data for Subframe 0 is recorded in the appropriate format (i.e. band interleaved by pixel, band interleaved by subframe, or band interleaved by line), followed by all the data for Subframe 1, and so on. In still another variant, the data in a given spectral band can be recorded for Subframes 0 through N, followed by the data for another spectral band for Subframes 0 through N, and so on. This is defined as band sequential format.

Band sequential and band interleaved formats are mutually exclusive: an image may be band sequential or band interleaved, but not both. Moreover, the various band interleaved formats are likewise mutually exclusive; an image may be band interleaved by pixel or by line or by subframe.

To provide the capability for RPF to handle each case, each [spatial data subsection] shall contain one or more [spectral group]s; within each [spectral group], each subframe may contain one or more [spectral band table]s; each [spectral band table] is composed of one or more [image row]s; each [image row] is composed of one or more [spectral band line]s; and each [spectral band line] is composed of one or more /image code/s.

In the band interleaved by pixel case, there will be one [spectral group]; within the [spectral group], each [subframe table] will contain one [spectral band table], and each [image row] will contain one [spectral band line]. Each pixel will comprise a series of data element values -- one for each spectral band represented.

In the band interleaved by line case, there will be one [spectral group]; within the [spectral group], each [subframe table] will contain one and only one [spectral band table], and each [image row] will contain multiple [spectral band line]s.

In the band interleaved by subframe case, there will be one [spectral group]; within the [spectral group], each [subframe table] will contain multiple [spectral band table]s, and each [image row] in each [spectral band table] will contain one and only one [spectral band line]. In the case of a subframe-level band sequential RGB image, three such tables would be used -- one each for red, green, and blue data.

In the band sequential case, there will be multiple [spectral group]s (one per band: three [spectral group]s in the RGB case); each [subframe table within each [spectral group] will contain one [spectral band table], and each [image row] will contain one [spectral band line].

The specification for a given product will specify the spectral band format of the spatial data.

5.1.12 Explicit areal coverage for attributes. Each [frame file] shall contain data that covers a given geographic region of the Earth. The [coverage section] in a given [frame file] shall define the four corners -- in terms of latitude and longitude -- of a rectangular boundary that completely contains all of the pixels constituting the frame. This rectangular boundary shall also represent the "implicit coverage" for attributes of the frame and their parameters. That is, if a separate *explicit* coverage for a given attribute parameter is not defined in the [attribute section], then the coverage for that parameter shall be equal to the boundary defined in the [coverage section].

If an explicit coverage is defined for one or more attributes (e.g. if the horizontal accuracy of a given frame is not a constant, but varies from one part of the frame to another), then each explicit coverage shall be expressed in the [attribute section] as the coordinates of a polygon that is entirely or partially contained in the given frame. That is, if a given attribute is associated with a given geographic region that overlaps parts of several frames, then the coordinates of the entire polygon can be recorded as the explicit coverage area in each of the frames that intersects the polygon.

5.2 Data structure standards. The specific data structures that conform to the RPF are described in the paragraphs below, including: [table of contents file]; [frame file]; and [external color/grayscale file].

5.2.1 Table of contents file.

a. The structure of the [table of contents file] will be compatible with the structure of the [frame file].

b. The overall structure shall be as follows:

[table of contents file]
{1}
[header section]
[location section]
[boundary rectangle section] (0, 1)
[frame file index section] (0, 1)
[colortable file index section] (0, 1)

(1) The [header section] will identify the file and provide security information. Note: the <location section location> field is intended to facilitate backward compatibility. See 5.1.8 above.)

(2) The [location section] will show the programmer the beginning byte locations (addresses) of the remaining sections in the file, relative to the beginning of the file.

(a) The programmer will be able to read in and interpret the [location section] and immediately know where to find the data of interest, and the length of each section.

(b) The [component location record]s in the [location section] will be stored in ascending sequence by <component id>.

(c) The <component aggregate length> field shall contain the sum of the <component length> fields in the [component location record]s. The receiver will be able to compare the <component aggregate length> with the sum of the individual <component length>s to ensure that the [location section] has been found correctly and to verify its structural integrity.

(3) The [boundary rectangle section] will contain the boundaries of one or more boundary rectangles, each defining the periphery of a geographic area containing all of the [frame file]s in the given data interchange volume that have a given data type, compression ratio, producer, latitudinal zone, and scale (or

resolution). A given record will also specify the dimensions of a rectangular "virtual matrix" of fixed-size frames of the given scale or resolution that fills the given boundary rectangle. The [frame file index section] will provide the identities of the subset of these frames that are actually recorded on the given interchange volume. If the [frame file index section] is omitted, then the [boundary rectangle section] shall be omitted. An example of a boundary rectangle is shown in FIGURE 1.

(4) The [frame file index section] will contain scales and data types for all [frame file]s in the given volume. Each entry will identify the boundary rectangle (named in the [boundary rectangle section]) where the frame is located, and it will specify the row and column in a "virtual matrix" of frames within the boundary rectangle where the specific frame is located. The information will enable the user to compute the coverage of the given frame, as specified in the corresponding [frame file].

The [pathname table] will show the pathname from the RPF directory of each [frame file] listed in the [frame file index table]. For example, for any [frame file] stored in directory RPF/CONC/CONCZ02, the pathname will be "./CONC/CONCZ02". Since the [pathname table] entries will vary in length, the [frame file index record] for a given [frame file] provides the offset and length of the <pathname> field in the [pathname table] that pertains to the given [frame file]. The [frame file]s stored in the same directory will have the same <pathname>. The [frame file index record]s for these [frame file]s will be able to use a single [pathname record].

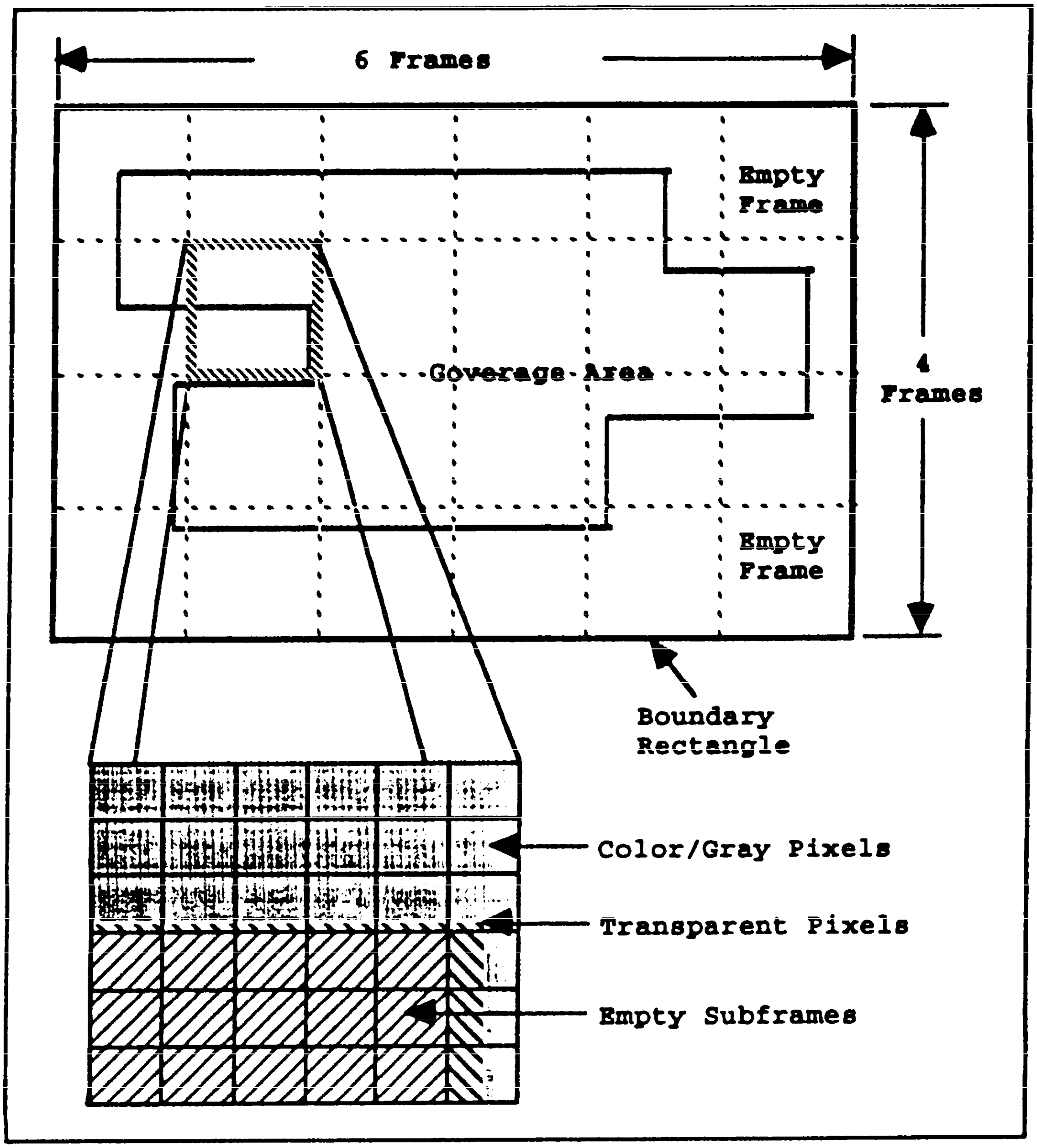

FIGURE 1. Example of a boundary rectangle.

(5) The [colortable file index section] will contain file names and brief narrative descriptions of each [external color/grayscale file]s, if any, stored in the given interchange volume.

c. The detailed structure of the [table of contents file] shall be as shown in FIGURE 2.

d. The [table of contents file] shall contain the following logical elements, listed in alphabetical order:

(1) <boundary rectangle record length> ::= a 2-byte unsigned integer ≥ 132 indicating the length in bytes of each [boundary rectangle record] (intended for backward compatibility; see 5.1.8 above).

(2) <boundary rectangle record number> ::= a 2-byte unsigned integer, 0 ≤ <boundary rectangle record number> ≤ <number of boundary rectangle records> - 1, defining the sequence number of the [boundary rectangle record] that defines the geographic area containing the region where the [frame file] identified in this [frame file index record] is located.

(3) <boundary rectangle table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [boundary rectangle subsection] and the first byte of the [boundary rectangle table] (counting the first byte of the [boundary rectangle subsection] as 0).

```
[table of contents file]
    {1}
    [header section]
        {2}
        <little/big endian indicator>,bool:1
        <header section length>,uint:2
        <file name>,asci:12
        <new/replacement/update indicator>,uint:1
        <governing standard number>,asci:15
        <governing standard date>,asci:8
        <security classification>,asci:1
        <security country/international code>,asci:2
        <security release marking>,asci:2
        <location section location>,uint:4
    {1}
    [location section]
        {2}
        <location section length>,uint:2
        <component location table offset>,uint:4
        <number of component location records>,uint:2
        <component location record length>,uint:2
        <component aggregate length>,uint:4
        [component location table]
            {3}
            [component location record] (2, ... many)
                {4}
                <component id>,uint:2
                <component length>,uint:4
                <component location>,uint:4
    {1}
    [boundary rectangle section] (0, 1)
        {2}
        [boundary rectangle section subheader]
            {3}
            <boundary rectangle table offset>,uint:4
            <number of boundary rectangle records>,uint:2
            <boundary rectangle record length>,uint:2
        {2}
        [boundary rectangle subsection]
            {3}
            [boundary rectangle table]
                {4}
                [boundary rectangle record] (1, ... many)
                    {5}
                    <product data type>,asci:5
                    <compression ratio>,asci:5
                    <scale or resolution>,asci:12
                    <zone>,asci:1
                    <producer>,asci:5
```

FIGURE 2. _[table of contents file]: detailed structure._

```
                    {5} (continued)
                    <northwest/upper left latitude>,real:8
                    <northwest/upper left longitude>,real:8
                    <southwest/lower left latitude>,real:8
                    <southwest/lower left longitude>,real:8
                    <northeast/upper right latitude>,real:8
                    <northeast/upper right longitude>,real:8
                    <southeast/lower right latitude>,real:8
                    <southeast/lower right longitude>,real:8
                    <north-south/vertical resolution>,real:8
                    <east-west/horizontal resolution>,real:8
                    <latitude/vertical interval>,real:8
                    <longitude/horizontal interval>,real:8
                    <number of frames in north-south or up-down
                        direction>,uint:4
                    <number of frames in east-west or left-
                        right direction>,uint:4
    {1}
    [frame file index section] (0, 1)
        {2}
        [frame file index section subheader]
            {3}
            <highest security classification>,asci:1
            <frame file index table offset>,uint:4
            <number of frame file index records>,uint:4
            <number of pathname records>,uint:2
            <frame file index record length>,uint:2
        {2}
        [frame file index subsection]
            {3}
            [frame file index table]
                {4}
                [frame file index record] (1, ... many)
                    {5}
                    <boundary rectangle record number>,uint:2
                    <frame location row number>,uint:2
                    <frame location column number>,uint:2
                <pathname record offset>,uint:4
                    <frame file name>,asci:12
                    <geographic location>,asci:6
                    <frame file security classification>,asci:1
                    <frame file security country/international
                        code>,asci:2
                    <frame file security release
                        marking>,asci:2
            {3}
            [pathname table]
                {4}
                [pathname record] (1, ... many)
                    {5}
                    <pathname length>,uint:2
                    <pathname>,asci:var
```

FIGURE 2. [table of contents file]: detailed structure -- Continued.

```
{1}
[colortable index section] (0, 1)
    {2}
    [colortable index section subheader]
        {3}
        <colortable index table offset>,uint:4
        <number of colortable index records>,uint:2
        <colortable index record length>,uint:2
    {2}
    [colortable index subsection]
        {3}
        [colortable index table]
            {4}
            [colortable index record] (1, .. many)
                {5}
                <external color/grayscale file
                  name>,asci:12
                <comment>,asci:50
```

FIGURE 2. <u>[table of contents file]: detailed structure</u> -- Concluded.

(4) <colortable index record length>,uint:2 ::= a 2-byte unsigned integer $\geq$ 62 indicating the length in bytes of each [colortable index record] (intended for backward compatibility; see 5.1.8 above).

(5) <colortable index table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [colortable index subsection] and the first byte of the [colortable index table] (counting the first byte of the [colortable index subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(6) <comment> ::= a 50-byte ASCII character string describing the characteristics of the [external color/grayscale file] named in this [colortable index record].

(7) <component aggregate length> ::= a 4-byte unsigned integer indicating the sum of the lengths in bytes of all of the <component length> fields in this [location section].

(8) <component id> ::= a 2-byte unsigned integer identifying a specific level-2 component of a given section (e.g. [compression parameter subsection] in the [compression section]) whose location is given in the corresponding <component location> field in this [component location record], encoded as defined in MIL-STD-2411-1, section 5.1.1.

(9) <component length>::= a 4-byte unsigned integer indicating the length in bytes of the component named in the corresponding <component id> field.

(10) <component location> ::= a 4-byte unsigned integer defining the absolute address (i.e. byte number) of the first byte of the component measured from the beginning of this RPF [frame file] (counting the first byte of the file as 0). If the file is encapsulated in a NITF message, the first byte is as specified in MIL-STD-2411-2.

(11) <component location record length> ::= a 2-byte unsigned integer ≥ 10 indicating the length in bytes of each [component location record]. (Intended for backward compatibility; see 5.1.8 above).

(12) <component location table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [location section] and the first byte of the [component location table] (counting the first byte of the [location section] as 0). (Intended for backward compatibility; see 5.1.8 above).

(13) <compression ratio> ::= a 5-byte ASCII character string encoded as specified in MIL-STD-2411-1, section 5.2.2, defining the nominal compression ratio (e.g. "55:1", "12:1", "UNC ") of the [frame file]s associated with this [boundary rectangle record] in human-readable form.

(14) <east-west/horizontal resolution> :: = an 8-byte real value specifying the nominal resolution, in the east-west or horizontal direction, for all data output pixel values derived from the data in the [frame file]s associated with the boundary rectangle identified in this [boundary rectangle record], measured in meters.

(15) <external color/grayscale file name> ::= a 12-byte ASCII character field in the format defined in 4.5.4.3 above (i.e. "XXXX.LUT   "), identifying an [external color/grayscale file] recorded in the RPF/LOOKUP directory on this volume.

(16) <file name> ::= a 12-byte ASCII character field ::= "A.TOC   " ::= the name of this [table of contents file], as specified in 4.5.4.2 above.

(17) <frame file index record length> ::= a 2-byte unsigned integer ≥ 33 indicating the length in bytes of each [boundary rectangle record]. (Intended for backward compatibility; see 5.1.8 above).

(18) <frame file index table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [frame file index subsection] and the

first byte of the [frame file index table] (counting the first byte of the [frame file index subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(19) <frame file name> ::= a 12-byte ASCII character field in the format defined in 4.5.4.4 above, identifying a [frame file] recorded on this volume.

(20) <frame file security classification> ::= a 1-byte ASCII character field coded to indicate the security level (e.g. unclassified, confidential, secret) of the indexed [frame file], as specified in MIL-STD-2411-1, section 5.1.8.

(21) <frame file security country/international code> ::= a 2-byte ASCII character string coded to indicate the originating country or international affiliation of the <security classification> of the indexed [frame file], as defined in MIL-STD-2411-1, section 5.1.7.

(22) <frame file security release marking> ::= a 2-byte ASCII character string coded to indicate any special handling or releasability restrictions assigned to the contents of the indexed [frame file], as defined in MIL-STD-2411-1, section 5.1.9.

(23) <frame location column number> ::= a 2-byte unsigned integer, 0 ≤ <frame location column number> ≤ <number of frames in east-west or left-right direction> - 1, defining the east-west or left-right coordinate of the [frame file] identified in this [frame file index record], within the virtual matrix of [frame file]s that comprise the corresponding [boundary rectangle record]. <frame location column number>s shall be assigned in ascending order beginning at the westernmost or leftmost edge of the boundary rectangle.

(24) <frame location row number> ::= a 2-byte unsigned integer, 0 ≤ <frame location row number> ≤ <number of frames in north-south or up-down direction> - 1, defining the north-south or up-down coordinate of the [frame file] identified in this [frame file index record], within the virtual matrix of [frame file]s that comprise the corresponding [boundary rectangle record]. <frame location row number>s shall be assigned in ascending order beginning at the southernmost or lowest edge of the boundary rectangle.

(25) <geographic location> ::= a 6-byte ASCII character string defining the geographic location of the southwest or lower left corner of the [frame file] identified in this [frame file index record] in GEOREF notation, encoded as specified in DMA TM 8358.1, section 5-4.

(26) <governing standard date> ::= an 8-byte ASCII character string in the format YYYYMMDD defining the effective

date of the standard (i.e. this document) to which the format of this [table of contents file] conforms.

(27) <governing standard number> ::= a 15-byte ASCII character string defining the document number of the standard (i.e. this document) to which the format of this [table of contents file] conforms.

(28) <header section length> ::= a 2-byte unsigned integer ≥ 48 indicating the length of the [header section] in bytes (intended for backward compatibility; see 5.1.8 above).

(29) <highest security classification> ::= a 1-byte ASCII character field indicating the highest security classification assigned to any [frame file] indexed in this [frame file index section], as defined in MIL-STD-2411-1.

(30) <latitude/vertical interval> ::= an 8-byte real value specifying the nominal latitude interval between adjacent decompressed pixels in the [frame file]s associated with the boundary rectangle identified in this [boundary rectangle record], measured in decimal degrees.

(31) <little/big endian indicator> ::= a 1-byte Boolean field ::= (FF)H to indicate that this [table of contents file] is recorded in little endian format, and ::= (00)H to indicate that this [table of contents file] is recorded in big endian format. See 4.4.1 above.

(32) <location section length> ::= a 2-byte unsigned integer ≥ 34 indicating the length in bytes of the entire [location section]

(33) <location section location> ::= a 4-byte unsigned integer defining the absolute address (i.e. byte number) of the first byte of the [location section] relative to the beginning of the file.

(34) <longitude/horizontal interval> ::= an 8-byte real value specifying the nominal longitude interval between adjacent output pixels in the [frame file]s associated with the boundary rectangle identified in this [boundary rectangle record], measured in decimal degrees.

(35) <new/replacement/update indicator> ::= a 1-byte unsigned integer ::= 0, indicating that this is a new [table of contents file].

(36) <north-south/vertical resolution> ::= an 8-byte real value specifying the nominal resolution, in the north-south or vertical direction, for all data output pixels derived from the data in the [frame file]s associated with the boundary rectangle

identified in this [boundary rectangle record], measured in meters.

(37) <northeast/upper right latitude> ::= an 8-byte real value corresponding to the latitude of the northeast or upper right corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(38) <northeast/upper right longitude> ::= an 8-byte real value corresponding to the longitude of the northeast or upper right corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(39) <northwest/upper left latitude> ::= an 8-byte real value corresponding to the latitude of the northwest or upper left corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(40) <northwest/upper left longitude> ::= an 8-byte real value corresponding to the longitude of the northwest or upper left corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(41) <number of boundary rectangle records> ::= a 2-byte unsigned integer ≥ 1 defining the number of [boundary rectangle record]s in this [boundary rectangle section].

(42) <number of colortable index records> ::= a 2-byte unsigned integer ≥ 1 defining the number of [colortable index record]s in this [colortable index section].

(43) <number of component location records> ::= a 2-byte unsigned integer ≥ 1 indicating the number of [component location record]s in this [location section].

(44) <number of frame file index records> ::= a 4-byte unsigned integer ≥ 1 defining the number of [frame file index record]s in this [frame file index section].

(45) <number of frames in east-west or left-right direction> ::= a 4-byte unsigned integer ≥ 1 specifying the number of frames in the east-west or left-right direction comprising the virtual geographic matrix of [frame file]s that are contained within the boundary rectangle whose corner coordinates are given in this [boundary rectangle record].

(46) <number of frames in north-south or up-down direction> ::= a 4-byte unsigned integer ≥ 1 specifying the number of frames in the north-south or up-down direction comprising the virtual geographic matrix of [frame file]s that are contained within the boundary rectangle whose corner coordinates are given in this [boundary rectangle record].

(47) <number of pathname records> ::= a 2-byte unsigned integer ≥ 1 indicating the number of [pathname record]s in the [pathname table].

(48) <pathname> ::= a variable-length ASCII character string defining the pathname for a given [frame file] (e.g. "./CONC/CONCZ02" for a [frame file] stored in the directory /RPF/CONC/CONCZ02).

(49) <pathname length> ::= a 2-byte unsigned integer ≥ 1 indicating the length in bytes of the <pathname> field in this [pathname record].

(50) <pathname record offset> ::= a 4-byte unsigned integer defining the address (i.e. byte number) of the first byte of the [pathname record] containing the <pathname> associated with this [frame file index record], relative to the beginning of the [frame file index subsection] (counting the first byte of the [frame file index subsection] as 0).

(51) <producer> ::= a 5-byte ASCII character string encoded as specified in MIL-STD-2411-1, section 5.2.1, identifying in human-readable form a designator for the organization (e.g. "DMAAC", "SOCAF", "AFESC") that produced the [frame file]s associated with this [boundary rectangle record].

(52) <product data type> ::= a 5-byte ASCII character string encoded as specified in MIL-STD-2411-1, section 5.1.6, defining the data type (e.g. "ADRG ", "DTED ") of the [frame file]s associated with this [boundary rectangle record] in human-readable form.

(53) <scale or resolution> ::= a 12-byte ASCII character string identifying in human-readable form the nominal scale (e.g. "1:1M", 1:12.5K") or nominal resolution (e.g. "100m" or "50m" that produced the [frame file]s associated with this [boundary rectangle record].

(54) <security classification> ::= a 1-byte ASCII character coded to indicate the security level (e.g. unclassified, confidential, secret) of this file, as specified in MIL-STD-2411-1, section 5.1.8.

(55) <security country/international code> ::= a 2-byte ASCII character string coded to indicate the originating country or international affiliation of the <security classification> of this file, as defined in MIL-STD-2411-1, section 5.1.7.

(56) <security release marking> ::= a 2-byte ASCII character string coded to indicate any special handling or releasability restrictions assigned to the contents of this file, as defined in MIL-STD-2411-1, section 5.1.9.

(57) <southeast/lower right latitude> ::= an 8-byte real value corresponding to the latitude of the southeast or lower right corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(58) <southeast/lower right longitude> ::= an 8-byte real value corresponding to the longitude of the southeast or lower right corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(59) <southwest/lower left latitude> ::= an 8-byte real value corresponding to the latitude of the southwest or lower left corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(60) <southwest/lower left longitude> ::= an 8-byte real value corresponding to the longitude of the southwest or lower left corner of the boundary rectangle defined in this [boundary rectangle record], measured in decimal degrees.

(61) <zone> ::= a 1-byte ASCII character string encoded as specified in MIL-STD-2411-1, section 5.1.3, identifying in human-readable form the latitudinal zone described in the [frame file]s associated with this [boundary rectangle record].

5.2.2 <u>Frame file</u>.

a. The overall structure of the [frame file] shall be as follows:

```
[frame file]
     {1}
     [header section]
     [location section]
     [coverage section] (0, 1)
     [compression section] (0, 1)
     [color/grayscale section] (0, 1)
     [image section]
     [attribute section] (0, 1)
     [related images section] (0, 1)
     [replace/update section] (0, 1)
```

(1) The [header section] will contain information that enables the programmer to uniquely identify the [frame file] and process it further. (Note: the <location section location> field is intended to facilitate backward compatibility. See 5.1.8 above.)

(2) The [location section] will show the programmer the beginning byte locations (addresses) of the remaining sections in the file, relative to the beginning of the file.

(a) The programmer will be able to read in and interpret the [location section] and immediately know where to find the data of interest, and calculate the length of each section.

(b) The [component location record]s in the [location section] will be stored in ascending sequence by <component id>.

(c) The <component aggregate length> field shall contain the sum of the <component length> fields in bytes in the [component location record]s. The receiver will be able to compare the <component aggregate length> with the sum of the individual <component length>s to ensure that the [location section] has been found correctly and to verify its structural integrity.

(3) If the [coverage section] is present it will describe the geographic coverage of the [frame file] in terms of latitude and longitude.

(4) The [compression section] will appear only in compressed files; it will contain algorithm identifiers, lookup tables, and other information that is specific to a given compression scheme. Each compression scheme, and the corresponding structure of the [compression section], will be described completely in the product specification for the compressed product.

(5) The [color/grayscale section] will appear only in [frame file]s that contain image or map data whose colors are not directly recorded in the [image section] of the given [frame file]. It will contain color and/or grayscale values for softcopy and hardcopy display, to be used for all the image data in the [frame file]. The [color/grayscale section] may contain multiple [color/grayscale table]s, and [color converter table]s that map the colors in one [color/grayscale table] into those of another.

(6) The [image section] will specify the compressed or uncompressed pixels that constitute the majority of the data volume in the file. Each frame will be tiled into one or more subframes for storage and processing.

(7) If the [attribute section] is present it will contain one or more attributes of the data described in the [image section], such as horizontal and vertical accuracy and datums. A listing of all registered attributes is given in MIL-STD-2411-1, section 5.3.2.

(a) The [attribute table] will show the <parameter value>s of each attribute of the given [frame file]. Since the [attribute table] entries will vary in length, the receiver may compute the length of the nth entry by subtracting the <parameter offset> address of the nth entry in the [attribute offset table] from the <parameter offset> address in the (n+1)th entry in the [attribute offset table].

(b) In the special case of the last <parameter value> in the table, the length shall be equal to:

length(last) = {<component location> for the [attribute subsection]} + {<component length> for the [attribute subsection]} - <parameter offset>(last) + 1,

where the <component location> and <component length> are found in the [component location record] for the [attribute subsection].

(8) If the [related images section] is present it will provide a list of [frame file]s on this volume that are related in a product-specific way to the image recorded in this [frame file]. For example, if a stereo pair of images are recorded, then each of the two images will be stored in a separate [frame file]; the [related images section]s of each will provide the file name and pathname of the other member of the pair.

(9) The [replace/update section] will identify previously produced [frame file]s that are replaced or updated by this [frame file]. A record will appear in the [replace/update table] for each previously produced [frame file] that is updated or superseded by the current [frame file]. The [replace/update section] shall appear only in [frame file]s that replace or update earlier editions. It shall not appear in the first complete edition of a given [frame file].

b. The detailed structure of the [frame file] shall be as shown in FIGURE 3.

c. The [frame file] shall contain the following logical elements, listed in alphabetical order:

(1) <areal coverage sequence number> ::= a 1-byte unsigned integer ≥ 0 that identifies the sequence number of an implicit or explicit areal coverage where the attribute defined in this [attribute offset record] applies. If <areal coverage sequence number> = 0, the given attribute applies (implicitly) to the entire geographic coverage of this [frame file]. If <areal coverage sequence number> > 0, then the areal coverage where the given attribute applies is defined in the corresponding [explicit areal coverage record] in the [explicit areal coverage table].

(2) <attribute id> ::= a 2-byte unsigned integer defining an attribute of the data in this [frame file], encoded as defined in MIL-STD-2411-1, section 5.3.2.

```
[frame file]
    {1}
    [header section]
        {2}
        <little/big endian indicator>,bool:1
        <header section length>,uint:2
        <file name>,asci:12
        <new/replacement/update indicator>,uint:1
        <governing specification number>,asci:15
        <governing specification date>,asci:8
        <security classification>,asci:1
        <security country/international code>,asci:2
        <security release marking>,asci:2
        <location section location>,uint:4
    {1}
    [location section]
        {2}
        <location section length>,uint:2
        <component location table offset>,uint:4
        <number of component location records>,uint:2
        <component location record length>,uint:2
        <component aggregate length>,uint:4
        [component location table]
            {3}
            [component location record] (3, ... many)
                {4}
                <component id>,uint:2
                <component length>,uint:4
                <component location>,uint:4
    {1}
    [coverage section] (0, 1)
        {2}
        [coverage section subheader]
            {3}
            <northwest/upper left latitude>,real:8
            <northwest/upper left longitude>,real:8
            <southwest/lower left latitude>,real:8
            <southwest/lower left longitude>,real:8
            <northeast/upper right latitude>,real:8
            <northeast/upper right longitude>,real:8
            <southeast/lower right latitude>,real:8
            <southeast/lower right longitude>,real:8
            <north-south/vertical resolution>,real:8
            <east-west/horizontal resolution>,real:8
            <latitude/vertical interval>,real:8
            <longitude/horizontal interval>,real:8
    {1}
    [compression section] (0, 1)
        {2}
        [compression section subheader]
```

FIGURE 3. [frame file]: detailed structure.

```
        {3}
        <compression algorithm id>,uint:2
        <number of compression lookup offset
          records>,uint:2
        <number of compression parameter offset
          records>,uint:2
    {2}
    [compression lookup subsection] (0, 1)
        {3}
        <compression lookup offset table offset>,uint:4
        <compression lookup table offset record
          length>,uint:2
        [compression lookup offset table]
            {4}
            [compression lookup offset record]
              (1, ... many)
                {5}
                <compression lookup table id>,uint:2
                <number of compression lookup
                  records>,uint:4
                <number of values per compression lookup
                  record>,uint:2
                <compression lookup value bit
                  length>,uint:2
                <compression lookup table offset>,uint:4
        {3}
        [compression lookup table] (1, ... many)
            {4}
            [compression lookup record] (1, ... many)
                {5}
                /compression lookup value/,bits:var
                  (1, ... many)
    {2}
    [compression parameter subsection] (0, 1)
        <compression parameter offset table offset>,uint:4
        <compression parameter offset record length>,uint:2
        {3}
        [compression parameter offset table]
            {4}
            [compression parameter offset record]
              (1, ... many)
                {5}
                <compression parameter id>,uint:2
                <compression parameter record
                  offset>,uint:4
        {3}
        [compression parameter record] (1, ... many)
            {4}
            <compression parameter value>,byte:var
```

FIGURE 3. [frame file]: detailed structure -- Continued.

```
{1}
[color/grayscale section] (0, 1)
    {2}
    [color/grayscale section subheader]
        {3}
        <number of color/grayscale offset records>,uint:1
        <number of color converter offset records>,uint:1
        <external color/grayscale file name>,asci:12
    {2}
    [colormap subsection] (0, 1)
        {3}
        <colormap offset table offset>,uint:4
        <color/grayscale offset record length>,uint:2
        [colormap offset table]
            {4}
            [color/grayscale offset record] (1, ... many)
                {5}
                <color/grayscale table id>,uint:2
                <number of color/grayscale records>,uint:4
                <color/grayscale element length>,uint:1
                <histogram record length>,uint:2
                <color/grayscale table offset>,uint:4
                <histogram table offset>,uint:4
        {3}
        [color/grayscale element group]
            {4}
            [color/grayscale table] (1, ... many)
                {5}
                [color/grayscale record] (1, ... many)
                    {6}
                    <color/grayscale element>,byte:var
        {3}
        [histogram element group] (0, 1)
            {4}
            [histogram table] (1, ... many)
                {5}
                [histogram record] (1, ... many)
                    {6}
                    <histogram element>,uint:4
    {2}
    [color converter subsection] (0, 1)
        {3}
        <color converter offset table offset>,uint:4
        <color converter offset record length>,uint:2
        <color converter record length>,uint:2
        [color converter offset table]
            {4}
            [color converter offset record] (1, ... many)
```

FIGURE 3. [frame file]: detailed structure -- Continued.

```
                    {5}
                    <color converter table id>,uint:2
                    <number of color converter records>, uint:4
                    <color converter table offset>,uint:4
                    <source color/grayscale offset table
                      offset>,uint:4
                    <target color/grayscale offset table
                      offset>,uint:4
                {4}
                [color converter table] (1, ... many)
                    {5}
                    [color converter record] (1, ... many)
                        {6}
                        <target color/grayscale table entry
                          number>,uint:4
    {1}
    [image section]
        {2}
        [image description subheader]
            {3}
            <number of spectral groups>,uint:2
            <number of subframe tables>,uint:2
            <number of spectral band tables>,uint:2
            <number of spectral band lines per image row>,uint:2
            <number of subframes in east-west or left-right
                direction>,uint:2
            <number of subframes in north-south or up-down
            direction>,uint:2
            <number of output columns per subframe>,uint:4
            <number of output rows per subframe>,uint:4
            <subframe mask table offset>,uint:4
            <transparency mask table offset>,uint:4
        {2}
        [mask subsection] (0, 1)
            {3}
            [mask subheader]
                {4}
                <subframe sequence record length>,uint:2
                <transparency sequence record length>,uint:2
                <transparent output pixel code length>,uint:2
                /transparent output pixel code/,bits:var (0, 1)
            {3}
            [subframe mask table] (0, 1)
                {4}
                [subframe mask spectral group] (1, ... many)
                {5}
                    [subframe mask row] (1, ... many)
                    {6}
                        [subframe sequence record](1, ... many)
                        {7}
                            <subframe offset>,uint:4
```

FIGURE 3. [frame file]: detailed structure -- Continued.

```
          {3}
          [transparency mask table] (0, 1)
               {4}
               [transparency mask spectral group] (1,... many)
               {5}
                    [transparency mask row] (1, ... many)
                    {6}
                         [transparency sequence record] (1, ... many)
                         {7}
                              <subframe offset>,uint:4
     {2}
     [image display parameters subheader]
          {3}
          <number of image rows>,uint:4
          <number of image codes per row>,uint:4
          <image code bit length>,uint:1
     {2}
     [spatial data subsection]
          {3}
          [spectral group] (1, ... many)
               {4}
               [subframe table] (1, ... many)
                    {5}
                    [spectral band table] (1, ... many)
                         {6}
                         [image row] (1, ... many)
                              {7}
                              [spectral band line] (1, ... many)
                                   {8}
                                   /image code/,bits:var
                                     (1, ... many)
{1}
[attribute section] (0, 1)
     {2}
     [attribute section subheader]
          {3}
          <number of attribute offset records>,uint:2
          <number of explicit areal coverage records>,uint:2
          <attribute offset table offset>,uint:4
          <attribute offset record length>,uint;2
     {2}
     [attribute subsection]
          {3}
          [attribute offset table]
               {4}
               [attribute offset record] (1, ... many)
```

FIGURE 3. [frame file]: detailed structure -- Continued.

```
                {5}
                <attribute id>,uint:2
                <parameter id>,uint:1
                <areal coverage sequence number>,uint:1
                <attribute record offset>,uint:4
        {3}
        [attribute table]
            {4}
            [attribute record] (1, ... many)
                {5}
                <parameter value>,byte:var (1, ... many)
    {2}
    [explicit areal coverage subsection] (0, 1)
        {3}
        <explicit areal coverage table offset>,uint:4
        <explicit areal coverage record length>,uint:2
        <corner coordinates record length>,uint:2
        [explicit areal coverage table]
            {4}
            [explicit areal coverage record] (1, ... many)
                {5}
                <number of vertices>,uint:2
                [corner coordinates record] (1, ... many)
                    {6}
                    <lat>,real:8
                    <long>,real:8
{1}
[related images section] (0, 1)
    {2}
    [related images section subheader]
        {3}
        <related image description table offset>,uint:4
        <number of related image description records>,uint:2
        <related image description record length>,uint:2
    {2}
    [related images subsection]
        {3}
        [related image description table]
            {4}
            [related image description record] (2, ... many)
                {5}
                <related image file name>,asci:12
                <related image pathname offset>,uint:4
                <relationship code>,uint:4
        {3}
        [related image pathname table]
```

FIGURE 3. [frame file]: detailed structure -- Continued.

```
                {4}
                [related image pathname record] (2, ... many)
                {5}
                    <related image pathname length>,uint:2
                    <related image pathname>,asci:var
    {1}
    [replace/update section] (0, 1)
        {2}
        [replace/update section subheader]
            {3}
            <replace/update table offset>,uint:4
            <number of replace/update records>,uint:2
            <replace/update record length>,uint:2
        {2}
        [replace/update subsection]
            {3}
            [replace/update table]
                {4}
                [replace/update record]  (1, ... many)
                    {5}
                    <new file name>,asci:12
                    <old file name>,asci:12
                    <replace/update status>,uint:1
```

FIGURE 3. [frame file]: detailed structure -- Concluded.

(3) <attribute offset record length> ::= a 2-byte unsigned integer ≥ 8 indicating the length in bytes of each [attribute offset record] (intended for backward compatibility; see 5.1.8 above).

(4) <attribute offset table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [attribute subsection] and the first byte of the [attribute offset table] (counting the first byte of the [attribute subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(5) <attribute record offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [attribute subsection] and the first byte of the [attribute record] whose <parameter value> field corresponds to the <parameter id> in this [attribute offset record] (counting the first byte of the [attribute subsection] as 0).

(6) <color converter offset record length> ::= a 2-byte unsigned integer ≥ 18 defining the length in bytes of each [color converter offset record] (intended for backward compatibility; see 5.1.8 above).

(7) <color converter offset table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [color converter subsection] and the first byte of the [color converter offset table] (counting the first byte of the [color converter subsection] as 0).

(8) <color converter record length> ::= a 2-byte unsigned integer defining the length in bytes of each [color converter record] in every [color converter table] (intended for backward compatibility; see 5.1.8 above)..

(9) <color converter table id> ::= a 2-byte unsigned integer defining the type of [color converter table] identified in this [color converter offset record], encoded as defined in MIL-STD-2411-1, section 5.1.2.2.

(10) <color converter table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [color converter subsection] and the first byte of the [color converter table] identified in this [color converter offset record] (counting the first byte of the [color converter subsection] as 0).

(11) <color/grayscale element> ::= a variable-length byte string (whose length is specified in <color/grayscale element length>) defining a unique color or grayscale value (or a combined color value and grayscale value) for the images in this [frame file]

(12) <color/grayscale element length> ::= a 1-byte unsigned integer defining the length in bytes of the <color/grayscale element> field in each [color/grayscale table].

(13) <color/grayscale offset record length> ::= a 2-byte unsigned integer ≥ 17 indicating the length in bytes of each [compression lookup table offset record] (intended for backward compatibility; see 5.1.8 above).

(14) <color/grayscale table id> ::= a 2-byte unsigned integer defining the type of color/grayscale table used with the image data in this [frame file], encoded as defined in MIL-STD-2411-1, section 5.1.2.1.

(15) <color/grayscale table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the [color/grayscale table] identified in this [color/grayscale offset record] (counting the first byte of the [colormap subsection] as 0).

(16) <colormap offset table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first

byte of the [colormap offset table] (counting the first byte of the [colormap subsection] as 0).

(17) <component aggregate length> ::= a 4-byte unsigned integer indicating the sum of the lengths in bytes of all of the <component length> fields in this [location section].

(18) <component id> ::= a 2-byte unsigned integer identifying a specific level-2 component of a given section (e.g. [compression parameter subsection] in the [compression section]) whose location is given in the corresponding <component location> field in this [component location record], encoded as defined in MIL-STD-2411-1, section 5.1.1.

(19) <component length>::= a 4-byte unsigned integer indicating the length in bytes of the component named in the corresponding <component id> field.

(20) <component location> ::= a 4-byte unsigned integer defining the absolute address (i.e. byte number) of the first byte of the component measured from the beginning of the file.

(21) <component location record length> ::= a 2-byte unsigned integer indicating the length in bytes of each [component location record].

(22) <component location table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [location section] and the first byte of the [component location table] (counting the first byte of the [location section] as 0). (Intended for backward compatibility; see 5.1.8 above).

(23) <compression algorithm id> ::= a 2-byte unsigned integer defining the compression algorithm used to decompress the image data in this [frame file], encoded as defined in MIL-STD-2411-1, section 5.3.1.

(24) <compression lookup offset table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [compression lookup subsection] and the first byte of the [compression lookup offset table] (counting the first byte of the [compression lookup subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(25) <compression lookup table id> ::= a 2-byte unsigned integer identifying the [lookup table] described in this [compression lookup table offset record], encoded as defined in MIL-STD-2411-1, section 5.3.1. The nth [compression lookup table offset record] shall contain the <compression lookup table id> of the nth [compression lookup table] in this [compression lookup subsection].

(26) <compression lookup table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [compression lookup subsection] and the first byte of the [compression lookup table] identified in this [compression lookup offset record] (counting the first byte of the [compression lookup subsection] as 0).

(27) <compression lookup table offset record length> ::= a 2-byte unsigned integer ≥ 14 indicating the length of each [compression lookup offset record] (intended for backward compatibility; see 5.1.8 above).

(28) /compression lookup value/ ::= a variable-length bit field defining an argument in a [compression lookup table] that is used in conjunction with a specific compression algorithm. The specification for a specific RPF data product that employs compression shall define the meaning and encoding of the /compression lookup value/ field.

(29) <compression lookup value bit length> ::= a 2-byte unsigned integer ≥ 4, defining the length in bits of the /compression lookup value/ field in each [compression lookup record] of each [compression lookup table] in the [compression section]. All /compression lookup value/ fields in a given [compression lookup table] shall have the same <compression lookup value bit length>, which shall be a multiple of 4 bits.

(30) <compression parameter id> ::= a 2-byte unsigned integer identifying a unique parameter that may be needed to interpret a compressed image file, encoded as defined in MIL-STD-2411-1, section 5.3.1.

(31) <compression parameter offset record length> ::= a 2-byte unsigned integer ≥ 8 indicating the length, measured in bytes, of each [compression parameter offset record] (intended for backward compatibility; see 5.1.8 above).

(32) <compression parameter offset table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [compression parameter subsection] and the first byte of the [compression parameter offset table] (counting the first byte of the [compression parameter subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(33) <compression parameter record offset> ::= a 4-byte unsigned integer defining the address (i.e. byte number) of a given [compression parameter record], corresponding to the <compression parameter id> in this [compression parameter offset record], relative to the beginning of this [compression parameter subsection] (counting the first byte of the [compression parameter subsection] as 0).

(34) <compression parameter value> ::= a variable-length field defining a parameter, associated with the compression algorithm identified in <compression algorithm id> and encoded as defined in MIL-STD-2411-1, section 5.3.1, that is needed to decompress the data in the [subframe subsection] in this [frame file].

(35) <corner coordinates record length> ::= a 2-byte unsigned integer ≥ 16 indicating the length, measured in bytes, of each [corner coordinates record] (intended for backward compatibility; see 5.1.8 above). If the [explicit areal coverage table] is omitted, then <corner coordinates record length> ::= a null value.

(36) <east-west/horizontal resolution> ::= an 8-byte real value specifying the nominal resolution, measured in meters, in the east-west or horizontal direction for all data output pixels derived from the /image code/s in [image section].

(37) <external color/grayscale file name> ::= a 12-byte alphanumeric ASCII character string formatted as specified in 4.5.4.3 above defining the name of a file containing a color/grayscale table for the image data recorded in this [frame file]. The file name shall consist of 8 bytes within the 12 byte field, padded with spaces.

(38) <explicit areal coverage record length> ::= a 2-byte unsigned integer indicating the length, measured in bytes, of each [explicit areal coverage record] in the [explicit areal coverage table].

(39) <explicit areal coverage table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [explicit areal coverage subsection] and the first byte of the [explicit areal coverage table] (counting the first byte of the [explicit areal coverage subsection] as 0).

(40) <file name> ::= a 12-byte ASCII character field defining the name of this [frame file], as specified in 4.5.4.4 above.

(41) <governing specification date> ::= an 8-byte ASCII character string in the format YYYYMMDD defining the effective date of the product specification to which the format of this [frame file] conforms.

(42) <governing specification number> ::= a 15-byte ASCII character string defining the document number of the product specification to which the format of this [frame file] conforms.

(43) <header section length> ::= a 2-byte unsigned integer ≥ 48 indicating the length of the [header section] in bytes (intended for backward compatibility; see 5.1.8 above).

(44) <histogram element> ::= A 4-byte unsigned integers, constituting a histogram. Each entry shall represent the absolute number of occurrences in the output pixel file of the corresponding <color/grayscale element>.

(45) <histogram record length> ::= a 2-byte unsigned integer ≥ 4 indicating the length in bytes of each [histogram record] (intended for backward compatibility; see 5.1.8 above). If the [histogram element group] is omitted, then <histogram record length> ::= a null value.

(46) <histogram table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the [histogram table] described in this [color/grayscale offset record] (counting the first byte of the [colormap subsection] as (0). If the [histogram element group] is omitted, then the <histogram table offset> ::= a null value.

(47) /image code/ ::= A bit string indicating one of the following:

(a) In an uncompressed map or image file, a fixed length pixel value.

(b) In a Vector Quantization compressed file, a fixed length index to a specific [row vector] in each [row record].

(c) A fixed-length or variable-length data code or specific parameter required for decompression as defined in the product specification.

Successive /image code/s shall be stored contiguously. If the total number of bits in the /image code/s constituting a given [spectral band table] is not a multiple of 8 bits, pad bits shall be inserted to insure the [spectral band table] consists of an integer number of bits. Each pad bit shall be ::= 1.

(48) <image code bit length> ::= a one-byte unsigned integer defining the length, in bits, of /image code/. An <image code bit length> of zero indicates a variable-length /image code/.

(49) <lat> ::= an 8-byte real value corresponding to the latitude of a vertex of a polygon in decimal degrees defining the areal extent of this attribute (see 5.1.12 above).

(50) <latitude/vertical interval> ::= an 8-byte real value specifying the nominal latitude interval between adjacent decompressed pixels, measured in decimal degrees.

(51) <little/big endian indicator> ::= a 1-byte Boolean field ::= (FF)H to indicate that this file is recorded in little

endian format, and ::= (00)H to indicate that this file is recorded in big endian format. See 4.4.1 above.

(52) <location section length> ::= a 2-byte unsigned integer ≥ 34 indicating the length in bytes of the entire [location section].

(53) <location section location> ::= a 4-byte unsigned integer defining the absolute address (i.e. byte number) of the first byte of the [location section] relative to the beginning of this RPF [frame file] (counting the first byte of the file as 0). If the file is encapsulated in a NITF message, the first byte is as specified in MIL-STD-2411-2.

(54) <long> ::= an 8-byte real value corresponding to the longitude of a vertex of a polygon in decimal degrees defining the areal extent of this attribute (see 5.1.12 above).

(55) <longitude/horizontal interval> ::= an 8-byte real value specifying the nominal longitude interval between adjacent decompressed pixels, measured in decimal degrees.

(56) <new file name> ::= a 12-byte ASCII character field, encoded as specified in 4.5.4.4. above, defining the file name of a [frame file] (i.e. the current [frame file] or one of its ancestors) that replaces or updates the [frame file] named in the <old file name> field in this [replace/update record].

(57) <new/replacement/update indicator> ::= a 1-byte unsigned integer encoded as follows:

::= 0 to indicate that this [frame file] contains original data that is not a replacement for a previous edition; an initial release of the data in this [frame file];

::= 1 to indicate that this [frame file] contains data that completely replaces all previous editions of the [frame file] for this data type, scale (or resolution), geographic coverage, and latitude zone.

::= 2 to indicate that this is an update (i.e. a patch) to a previous edition of this [frame file] with the same data type, scale (or resolution), geographic coverage, and latitude zone.

(58) <north-south/vertical resolution> ::= a 8-byte real value specifying the nominal resolution, measured in meters, in the north-south or vertical direction for all decompressed pixels represented in /image code/.

(59) <northeast/upper right latitude> ::= an 8-byte real value corresponding to the latitude of the northeast corner

(upper right corner in an RPF [frame file] that is not oriented north-up, such as a polar ADRG map) of the frame.

(60) <northeast/upper right longitude> ::= an 8-byte real value corresponding to the longitude of the northeast corner (upper right corner in an RPF [frame file] that is not oriented north-up, such as a polar ADRG map) of the frame.

(61) <northwest/upper left latitude> ::= an 8-byte real value corresponding to the latitude of the northwest corner (upper left corner in an RPF [frame file] that is not oriented north-up, such as a polar ADRG map) of the frame.

(62) <northwest/upper left longitude> ::= an 8-byte real value corresponding to the longitude of the northwest corner (upper left corner in an RPF [frame file] that is not oriented north-up, such as a polar ADRG map) of the frame.

(63) <number of attribute offset records> ::= a 2-byte unsigned integer defining the number of [attribute offset record]s in this [attribute section].

(64) <number of color/grayscale offset records> ::= a 1-byte unsigned integer ≥ 1, indicating the number of [color/grayscale offset record]s in the [colormap offset table].

(65) <number of color/grayscale records> ::= a 4-byte unsigned integer ≥ 1, indicating the number of [color/grayscale record]s in the [color/grayscale table] identified in this [color/grayscale offset record] contains, and the number of [histogram record]s the [histogram table] contains.

(66) <number of color converter offset records> ::= a 1-byte unsigned integer indicating the number of [color converter offset record]s in the [color converter offset table].

(67) <number of color converter records> ::= a 4-byte unsigned integer ≥ 1, indicating the number of [color converter record]s in the [color converter table] identified by the <color converter table offset> in this [color converter offset record]. The <number of color converter records> in a given [color converter offset record] shall equal the <number of color/grayscale records> in the source [color/grayscale table], identified by the <source color/grayscale table offset> in this [color converter offset record].

(68) <number of component location records> ::= a 2-byte unsigned integer indicating the number of [component location record]s in this [location section].

(69) <number of compression lookup records> ::= a 4-byte unsigned integer ≥ 1, indicating the number of [compression lookup record]s in each [compression lookup table].

(70) <number of compression lookup offset records> ::= a 2-byte unsigned integer ≥ 1, indicating the number of [compression lookup offset record]s in the [compression lookup offset table].

(71) <number of compression parameter offset records> ::= a 2-byte unsigned integer ≥ 0, indicating the number of [compression parameter offset record]s in the [compression parameter subsection].

(72) <number of explicit areal coverage records> ::= a 2-byte unsigned integer specifying the number of explicit areal coverage records in the [explicit areal coverage table].

(73) <number of image codes per row> ::= a 4-byte unsigned integer, defining the number of /image code/ fields in each [image row] of each [spectral band table]. A <number of image codes per row> of zero indicates a variable number of /image code/s per image row.

(74) <number of image rows> ::= a 4-byte unsigned integer ≥ 1, indicating the number of [image row]s in each [spectral band table]. All [spectral band table]s in every [subframe table] shall contain the same number of [image row]s.

(75) <number of output columns per subframe> ::= a 4-byte unsigned integer specifying the number of columns in the output array of pixels. All subframes in a given [frame file] shall have the same number of columns in the output pixel array.

(76) <number of output rows per subframe> ::= a 4-byte unsigned integer specifying the number of rows in the output array of pixels. All subframes in a given [image section] shall have the same number of rows in the output pixel array.

(77) <number of related image description records> ::= a 2-byte unsigned integer indicating the number of [related image description record]s in the [related image description table].

(78) <number of replace/update records> ::= a 2-byte unsigned integer ≥ 1, indicating the number of [replace/update record]s in this [replace/update section].

(79) <number of spectral band lines per image row> ::= a 2-byte unsigned integer ≥ 1, defining the number of spectral bands that constitute an image row within each [subframe table] in this [image section]. The value shall equal 1 in images recorded in interleaved and band sequential format (see 5.1.11 above); the value shall be > 1 in an image stored in band interleaved by line format (e.g. the value = 3 in an RGB image stored in band interleaved by line format).

(80) <number of spectral band tables> ::= a 2-byte unsigned integer indicating the number of [spectral band table]s in each [subframe table] in this [image section].

(81) <number of spectral groups> ::= a 2-byte unsigned integer indicating the number of [spectral group]s in this [image section].

(82) <number of subframe tables> ::= a 2-byte unsigned integer ≥ 1, indicating the actual number of [subframe table]s in this [image section].

(83) <number of subframes in east-west or left-right direction> ::= a 2-byte unsigned integer defining the number of columns in the rectangular geographic array of subframes constituting the frame that is the subject of this [frame file].

(84) <number of subframes in north-south or up-down direction> ::= a 2-byte unsigned integer defining the number of rows in the rectangular geographic array of subframes constituting the frame that is the subject of this [frame file].

(85) <number of values per compression lookup record> ::= a 2-byte unsigned integer ≥ 1, indicating the number of contiguous /compression lookup value/ fields in each [compression lookup record] of a given [compression lookup table].

(86) <number of vertices> ::= a 2-byte unsigned integer specifying the number of <lat>, <long> pairs associated with a given explicit areal coverage. (See 5.1.12 above.)

(87) <old file name> ::= a 12-byte ASCII character string indicating the name of a previous [frame file], encoded as specified in 4.5.4.4 above, which is replaced or updated by the [frame file] identified in the <new file name> field> in this [replace/update record].

(88) <parameter id> ::= a 1-byte unsigned integer defining a specific parameter type, such as horizontal or vertical accuracy, horizontal datum code, etc. whose value is given in the corresponding <parameter value> field and whose relative address is given in the corresponding <parameter offset> field. The registered values for <parameter id> can be found in MIL-STD-2411-1, Section 5.3.2.

(89) <parameter value> ::= a variable-length field whose coding depends on the parameter type, defined in the corresponding <parameter id> field, encoded as specified in MIL-STD-2411-1, section 5.3.2. See the discussion of <parameter value> in 5.2.2.a.(8)(a) and 5.2.2.a(8)(b).

(90) <related image description record length> ::= a 2-byte unsigned integer ≥ 20 indicating the length in bytes of each

[related image description record] (intended for backward compatibility; see 5.1.8 above).

(91) <related image description table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [related images subsection] and the first byte of the [related image description table] (counting the first byte of the [related images subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(92) <related image file name> ::= a 12-byte ASCII character string in the format defined in 4.5.4.4 above, identifying a [frame file] recorded on this volume that is related to this [frame file] in a manner defined by the <relationship code> in this [related image description record].

(93) <related image pathname> ::= a variable-length ASCII character string defining the pathname for the directory containing a given [frame file], recorded on this volume for an image that is related to the image recorded in this [frame file], in accordance with the relationship defined in the corresponding <relationship code> in the [related image description record] for the related file. For example, if the image in [frame file] /RPF/CONC/CONCZ03/EFGH5678.ON3 is related to the image in this [frame file], then <related image pathname> ::= "./CONC/CONCZ03" in the [related image pathname record], and the corresponding <related image file name> ::= "EFGH5678.ON3" in the appropriate [related image description record].

(94) <related image pathname length> ::= a 2-byte unsigned integer indicating the length in bytes of the <related image pathname> field for the [frame file] identified in this [related image description record].

(95) <related image pathname offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [related images subsection] and the first byte of the record in the [related image pathname table] containing the pathname of the image file named in this [related image description record](counting the first byte of the [related images subsection] as 0).

(96) <relationship code> ::= a 4-byte unsigned integer defining the nature of the relationship (e.g. stereo pairs) between this [frame file] and the [frame file] identified in this [related image description record], encoded as specified in MIL-STD-2411-1, section 5.3.3.

(97) <replace/update record length> ::= a 2-byte unsigned integer ≥ 25 indicating the length in bytes of each [replace/update record] (intended for backward compatibility; see 5.1.8 above).

(98) <replace/update status> ::= a 1-byte unsigned integer indicating the status of this [replace/update record], encoded as follows:

::= 1 to indicate that the [frame file] identified in <new file name> field in this [replace/update record] updates one or more subframes in the [frame file] identified in the <old file name> field in this [replace/update record].

::= 2 to indicate that the [frame file] identified in <new file name> (which is intended to update a parent edition supersedes the [frame file] identified in the <old file name> field in this [replace/update record] (which also was intended to update the same parent edition).

::= 3 to indicate that the [frame file] identified in <new file name> field in this [replace/update record] replaces the entire [frame file] identified in the <old file name> field in this [replace/update record].

(99) <replace/update table offset> ::= a 4-byte integer defining the displacement, measured in bytes, between the beginning of the [replace/update subsection] and the first byte of the [replace/update table], (counting the first byte of the [replace/update subsection] as 0).

(100) <security classification> ::= a 1-byte ASCII character coded to indicate the security level (e.g. unclassified, confidential, secret) of this file, as specified in MIL-STD-2411-1, section 5.1.8.

(101) <security country/international code> ::= a 2-byte ASCII character string coded to indicate the originating country or international affiliation of the <security classification> of this file, as defined in MIL-STD-2411-1, section 5.1.7.

(102) <security release marking> ::= a 2-byte ASCII character string coded to indicate any special handling or releasability restrictions assigned to the contents of this file, as defined in MIL-STD-2411-1, section 5.1.9.

(103) <source color/grayscale offset table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the source [color/grayscale offset table], (counting the first byte of the [colormap subsection] as 0). The source [color/grayscale offset table] shall be the table corresponding to the [color/grayscale table] whose color entries are to be converted into those of the target [color/grayscale table], whose corresponding [color/grayscale offset table] is identified by the <target color/grayscale offset table offset> in this [color converter offset record].

(104) <southeast/lower right latitude> ::= an 8-byte real value corresponding to the latitude in decimal degrees of the southeast or lower right corner of the frame.

(105) <southeast/lower right longitude> ::= an 8-byte real value corresponding to the longitude in decimal degrees of the southeast or lower right corner of the frame.

(106) <southwest/lower left latitude> ::= an 8-byte real value corresponding to the latitude in decimal degrees of the southwest or lower left corner of the frame.

(107) <southwest/lower left longitude> ::= an 8-byte real value corresponding to the longitude in decimal degrees of the southwest or lower left corner of the frame.

(108) <subframe mask table offset> ::= a 4-byte unsigned integer indicating one of the following:

(a) the displacement, measured in bytes, between the beginning of the [mask subsection] and the first byte of the [subframe mask table] (counting the first byte of the [mask subsection] as 0).

(b) a null value in the case where the [subframe mask table] is not recorded. A null value will be recorded if no [mask subsection] is present.

(109) <subframe offset> ::= a 4-byte unsigned integer indicating one of the following:

(a) the displacement, measured in bytes, between the beginning of the [spatial data subsection] and the first byte of the [subframe table] for the subframe identified in this [subframe sequence record] or [transparency sequence record] (counting the first byte of the [spatial data subsection] as 0); or

(b) a null value in the case where the [subframe table] for the subframe identified in this [subframe sequence record] or [transparency sequence record] is not recorded.

(110) <subframe sequence record length> ::= a 2-byte unsigned integer ≥ 2 indicating the length in bytes of each [subframe sequence record] (intended for backward compatibility; see 5.1.8 above). If the [subframe mask table] is omitted, then <subframe sequence record length> ::= 0.

(111) <target color/grayscale table entry number> ::= a 4-byte unsigned integer ≤ the <number of color/grayscale records> - 1 in the target [color/grayscale table], indicating an entry number in the target [color/grayscale table]. The <target color/grayscale table entry number> in the nth [color converter record] shall relate the nth entry in the source [color/grayscale table] with the entry defined by the given <target color/grayscale table entry number> in the target [color/grayscale table]. The source and target [color/grayscale offset table]s for the source and target [color/grayscale table]s shall be identified in the [color converter offset record] for this [color converter table].

(112) <target color/grayscale offset table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the target [color/grayscale offset table], (counting the first byte of the [colormap subsection] as 0). The target [color/grayscale offset table] shall be the table corresponding to the [color/grayscale table] whose color entries are to be used in place of those of the source [color/grayscale table], whose corresponding [color/grayscale offset table] is identified by the <source color/grayscale offset table offset> in this [color converter offset record].

(113) <transparency mask table offset> ::= a 4-byte unsigned integer indicating one of the following:

(a) the displacement, measured in bytes, between the beginning of the [mask subsection] and the first byte of the [transparency mask table] (counting the first byte of the [mask subsection] as 0). (Intended for backward compatibility; see 5.1.8 above).

(b) a null value in the case where the [transparency mask table] is not recorded.

(114) <transparency sequence record length> ::= a 2-byte unsigned integer ≥ 2 indicating the length in bytes of each [transparency sequence record] (intended for backward compatibility; see 5.1.8 above). If the [transparency mask table] is omitted, then <transparency sequence record length> ::= 0.

(115) /transparent output pixel code/ ::= a variable-length bit string indicating the data value that shall be interpreted as follows:

(a) The value of /image code/ that corresponds to a transparent pixel or transparent color index value in a [frame file] containing uncompressed spatial data;

(b) the data value in the decompressed output file that corresponds to a transparent pixel, or a transparent color index value that the receiver shall translate into a transparent pixel, in a [frame file] containing compressed spatial data.

The value and interpretation of this field shall be defined in product specifications for individual RPF-compatible data products. If <transparent output pixel code length> ::= 0, then the /transparent pixel output code/ shall not be recorded.

(116) <transparent output pixel code length> ::= a 2-byte unsigned integer indicating the length, measured in bits, of the /transparent output pixel code/ field. The value and interpretation of this field shall be defined in product specifications for individual RPF-compatible data products.

5.2.3 External color/grayscale file.

a. The structure of the [external color/grayscale file] will be compatible with the structure of the [frame file]. If a color/grayscale table] is associated with a given [frame file], the table may be stored in the [frame file] itself or in an [external color/grayscale file]. An [external color/grayscale file] could be used for example, to house a very large colortable that would otherwise add an unacceptable overhead burden and significantly reduce the effective compression ratio if it were included in individual [frame file]s. In this case, it would be appropriate to store a single colortable in an [external color/grayscale file] and use it in conjunction with a family of images on an interchange volume.

For instance, if an 8-bit (256-entry) colortable with a 3-byte argument (768 bytes all told) is included in a compressed raster file of 280,000 bytes, it adds less than 0.3% to the total file size; however, if a 16-bit (65,536-entry) colortable with a 3-byte argument (196,608 bytes all told) is added to the same [frame file], it increases the file size by 70%. This situation would dictate use of an [external color/grayscale file].

b. The overall structure shall be as follows:

```
[external color/grayscale file]
    {1}
    [header section]
    [location section]
    [color/grayscale section]
```

(1) The [header section] will contain information that enables the programmer to uniquely identify the [external color/grayscale file] and process it further. Note: the

<location section location> field is intended to facilitate backward compatibility. See 5.1.8 above.)

(2) The [location section] will show the programmer the beginning byte locations (addresses) of the remaining sections in the file, relative to the beginning of the [nitf message] that encapsulates this file, as defined in MIL-STD-2411-2.

(a) The programmer will be able to read in and interpret the [location section] and immediately know where to find the data of interest, and calculate the length of each section.

(b) The [component location record]s in the [location section] will be stored in ascending sequence by <component id>.

(c) The <component aggregate length> field shall contain the sum of the <component length> fields in bytes in the [component location record]s. The receiver will be able to compare the <component aggregate length> with the sum of the individual <component length>s to ensure that the [location section] has been found correctly and to verify its structural integrity.

(3) The [color/grayscale section] will contain color and/or grayscale values for softcopy and hardcopy display, to be used for all the image data in a given [frame file].

(a) A [frame file] that uses an external [color/grayscale table] will provide the name of the corresponding [external color/grayscale file] and will not incorporate its own internal [color/grayscale table].

(b) The [color/grayscale section] may contain multiple [color/grayscale table]s, and [color converter table]s that map the colors in one [color/grayscale table] into those of another.

(c) The [histogram table] is optional. If the [histogram subsection] is omitted, then each [frame file] that references this [external color/grayscale file] may furnish its own internal [histogram table].

c. The detailed structure of the [external color/grayscale file] shall be as shown in FIGURE 4.

d. The [external color/grayscale file] shall contain the following fields, listed in alphabetical order:

(1) <color converter offset record length> ::= a 2-byte unsigned integer ≥ 18 defining the length in bytes of each [color converter offset record] (intended for backward compatibility; see 5.1.8 above)..

(2) <color converter record length> ::= a 2-byte unsigned integer defining the length in bytes of each [color converter record] in every [color converter table] (intended for backward compatibility; see 5.1.8 above)..

(3) <color converter offset table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [color converter subsection] and the first byte of the [color converter table] (counting the first byte of the [color converter subsection] as 0).

(4) <color converter table id> ::= a 2-byte unsigned integer defining the type of [color converter table] identified in this [color converter offset record], encoded as defined in MIL-STD-2411-1, section 5.1.2.2.

(5) <color converter table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [color converter subsection] and the first byte of the [color converter table] identified in this [color converter offset record] (counting the first byte of the [color converter subsection] as 0).

(6) <color/grayscale element> ::= a variable-length byte string (whose length is specified in <color/grayscale element length>) defining a unique color or grayscale value (or a combined color value and grayscale value) for the images in the [frame file]s that use this [external color/grayscale file].

(7) <color/grayscale element length> ::= a 1-byte unsigned integer defining the length in bytes of the <color/grayscale element> field in each [color/grayscale table].

(8) <color/grayscale offset record length> ::= a 2-byte unsigned integer ≥ 17 indicating the length of each [color/grayscale offset record] (intended for backward compatibility; see 5.1.8 above).

(9) <color/grayscale table id> ::= a 2-byte unsigned integer defining the color/grayscale table used with the image data in this [frame file], encoded as defined in MIL-STD-2411-1, section 5.1.2.1.

(10) <color/grayscale table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the [color/grayscale table] identified in this [color/grayscale offset record] (counting the first byte of the [colormap subsection] as 0).

```
[external color/grayscale file]
    {1}
    [header section]
        {2}
        <little/big endian indicator>,bool:1
        <header section length>,uint:2
        <file name>,asci:12
        <new/replacement/update indicator>,uint:1
        <governing specification number>,asci:15
        <governing specification date>,asci:8
        <security classification>,asci:1
        <security country/international code>,asci:2
        <security release marking>,asci:2
        <location section location>,uint:4
    {1}
    [location section]
        {2}
        <location section length>,uint:2
        <component location table offset>,uint:4
        <number of component location records>,uint:2
        <component location record length>,uint:2
        <component aggregate length>,uint:4
        [component location table]
            {3}
            [component location record] (2, ... many)
                {4}
                <component id>,uint:2
                <component length>,uint:4
                <component location>,uint:4
    {1}
    [color/grayscale section] (0, 1)
        {2}
        [color/grayscale section subheader]
            {3}
            <number of color/grayscale offset records>,uint:1
            <number of color converter offset records>,uint:4
            <dummy file name>,asci:12
        {2}
        [colormap subsection] (0, 1)
            {3}
            <colormap offset table offset>,uint:4
            <color/grayscale offset record length>,uint:2
            [colormap offset table]
                {4}
                [color/grayscale offset record] (1, ... many)
                    {5}
                    <color/grayscale table id>,uint:2
                    <number of color/grayscale records>,uint:4
```

FIGURE 4: Structure of the [external color/grayscale file].

```
                 {5} (continued)
                 <color/grayscale element length>,uint:1
                 <histogram record length>,uint:2
                 <color/grayscale table offset>,uint:4
                 <histogram table offset>,uint:4
         {3}
         [color/grayscale element group]
             {4}
             [color/grayscale table] (1, ... many)
                 {5}
                 [color/grayscale record] (1, ... many)
                     {6}
                     <color/grayscale element>,byte:var
         {3}
         [histogram element group] (0, 1)
             {4}
             [histogram table] (1, ... many)
                 {5}
                 [histogram record] (1, ... many)
                     {6}
                     <histogram element>,uint:4
     {2}
     [color converter subsection] (0, 1)
         {3}
         <color converter offset table offset>,uint:4
         <color converter offset record length>,uint:2
         <color converter record length>,uint:2
         [color converter offset table]
             {4}
             [color converter offset record] (1, ... many)
                 {5}
                 <color converter table id>,uint:2
                 <number of color converter records>,uint:4
                 <color converter table offset>,uint:4
                 <source color/grayscale offset table
                   offset>,uint:4
                 <target color/grayscale offset table
                   offset>,uint:4
             {4}
             [color converter table] (0, ... many)
                 {5}
                 [color converter record] (1, ... many)
                     {6}
                     <target color/grayscale table entry
                       number>,uint:4
```

FIGURE 4: <u>Structure of the [external color/grayscale file]</u> -- Concluded.

(11) <colormap offset table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the [colormap offset table] (counting the first byte of the [colormap subsection] as 0).

(12) <component aggregate length> ::= a 4-byte unsigned integer indicating the sum of the lengths in bytes of all of the <component length> fields in this [location section].

(13) <component id> ::= a 2-byte unsigned integer identifying a specific level-2 component of a given section (e.g. [compression parameter subsection] in the [compression section]) whose location is given in the corresponding <component location> field in this [component location record], encoded as defined in MIL-STD-2411-1, section 5.1.1.

(14) <component length>::= a 4-byte unsigned integer indicating the length in bytes of the component named in the corresponding <component id> field.

(15) <component location> ::= a 4-byte unsigned integer defining the absolute address (i.e. byte number) of the first byte of the [location section] measured from the relative to the beginning of this RPF [frame file] (counting the first byte of the file as 0). If the file is encapsulated in a NITF message, the first byte is as specified in MIL-STD-2411-2.

(16) <component location record length> ::= a 2-byte unsigned integer indicating the length in bytes of each [component location record].

(17) <component location table offset> ::= a 4-byte unsigned integer indicating the displacement, measured in bytes, between the beginning of the [location section] and the first byte of the [component location table] (counting the first byte of the [location section] as 0). (Intended for backward compatibility; see 5.1.8 above).

(18) <dummy file name> ::= a 12-byte alphanumeric ASCII character string ::= a null value. This field is specified to maintain consistency between the structure of the [color/grayscale section] of the [external color/grayscale file] and that of the [color/grayscale section] of the [frame file]. The corresponding field in the [frame file] provides the name of the related [external color/grayscale file]; the concept of a related [external color/grayscale file] does not exist in the [external color/grayscale file] itself, so this field shall be null.

(19) <file name> ::= a 12-byte ASCII character field indicating the name of this [external color/grayscale file], encoded as specified in 4.5.4.3 above. The file name shall

consist of 8 bytes within the 12 byte field, padded with spaces in the rightmost four characters.

(20) <governing specification date> ::= an 8-byte ASCII character string in the format YYYYMMDD defining the effective date of the product specification to which the format of this [frame file] conforms.

(21) <governing specification number> ::= a 15-byte ASCII character string defining the document number of the product specification to which the format of this [frame file] conforms.

(22) <header section length> ::= a 2-byte unsigned integer ≥ 48 indicating the length of the [header section] in bytes (intended for backward compatibility; see 5.1.8 above).

(23) <histogram element> ::= 4-byte unsigned integer, constituting a histogram. Each entry shall represent the relative frequency of occurrence in the output pixel file of the corresponding <color/grayscale element>.

(24) <histogram record length> ::= a 2-byte unsigned integer ≥ 4 indicating the length in bytes of each [histogram record] (intended for backward compatibility; see 5.1.8 above). If the [histogram element group] is omitted, then <histogram record length> ::= a null value.

(25) <histogram table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the [histogram table] described in this [color/grayscale offset record] (counting the first byte of the [colormap subsection] as (0). If the [histogram element group] is omitted, then the <histogram table offset> ::= a null value.

(26) <little/big endian indicator> ::= a 1-byte Boolean field ::= (FF)H to indicate that this file is recorded in little endian format, and ::= (00)H to indicate that this file is recorded in big endian format.

(27) <location section length> ::= a 2-byte unsigned integer ≥ 34 indicating the length in bytes of the entire [location section].

(28) <location section location> ::= a 4-byte unsigned integer defining the absolute address (i.e. byte number) of the first byte of the [location section] relative to the beginning of relative to the beginning of this RPF [frame file] (counting the first byte of the file as 0). If the file is encapsulated in a NITF message, the first byte is as specified in MIL-STD-2411-2.

(29) <new/replacement/update indicator> ::= a 1-byte unsigned integer ::= 0 to indicate that this is a new [external color/grayscale file].

(30) <number of color converter offset records> ::= a 4-byte unsigned integer indicating the number of [color converter offset record]s in the [color converter offset table].

(31) <number of color converter records> ::= a 4-byte unsigned integer ≥ 1, indicating the number of [color converter record]s in the [color converter table] identified by the <color converter table offset> in this [color converter offset record]. The <number of color converter records> in a given [color converter offset record] shall equal the <number of color/grayscale records> in the source [color/grayscale table], identified by the <source color/grayscale table offset> in this [color converter offset record].

(32) <number of color/grayscale offset records> ::= a 1-byte unsigned integer ≥ 1, indicating the number of [color/grayscale offset record]s in the [colormap offset table].

(33) <number of color/grayscale records> ::= a 4-byte unsigned integer ≥ 1, indicating the number of [color/grayscale record]s the [color/grayscale table] identified in this [color/grayscale offset record] contains, and the number of [histogram record]s the [histogram table] contains.

(34) <number of component location records> ::= a 2-byte unsigned integer indicating the number of [component location record]s in this [location section].

(35) <security classification> ::= a 1-byte ASCII character coded to indicate the security level (e.g. unclassified, confidential, secret) of this file, as specified in MIL-STD-2411-1, section 5.1.8.

(36) <security country/international code> ::= a 2-byte ASCII character string coded to indicate the originating country or international affiliation of the <security classification> of this file, as defined in MIL-STD-2411-1, section 5.1.7.

(37) <security release marking> ::= a 2-byte ASCII character string coded to indicate any special handling or releasability restrictions assigned to the contents of this file, as defined in MIL-STD-2411-1, section 5.1.9.

(38) <source color/grayscale offset table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the source [color/grayscale offset table], (counting the first byte of the [colormap subsection] as 0). The source [color/grayscale offset table] shall be the table corresponding to the [color/grayscale table] whose color entries are to be converted into those of the target [color/grayscale table], whose corresponding [color/grayscale offset table] is identified by the <target color/grayscale offset table offset> in this [color converter offset record].

(39) <target color/grayscale table entry number> ::= a 4-byte unsigned integer ≤ the <number of color/grayscale records> - 1 in the target [color/grayscale table], indicating an entry number in the target [color/grayscale table]. The <target color/grayscale table entry number> in the nth [color converter record] shall relate the nth entry in the source [color/grayscale table] with the entry defined by the given <target color/grayscale table entry number> in the target [color/grayscale table]. The source and target [color/grayscale offset table]s for the source and target [color/grayscale table]s shall be identified in the [color converter offset record] for this [color converter table].

(40) <target color/grayscale offset table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [colormap subsection] and the first byte of the target [color/grayscale offset table], (counting the first byte of the [colormap subsection] as 0). The target [color/grayscale offset table] shall be the table corresponding to the [color/grayscale table] whose color entries are to be used in place of those of the source [color/grayscale table], whose corresponding [color/grayscale offset table] is identified by the <source color/grayscale offset table offset> in this [color converter offset record].

6. NOTES

(This section contains information of a general or explanatory nature that may be helpful, but is not mandatory.)

6.1 Intended use. This standard is intended to govern the standard and design of a family of digital data interchange products that comprise raster maps, images, and other geographic data for military applications.

6.2 Acquisition requirements. When this specification is used in acquisition, the applicable issue of the DODISS must be cited in the solicitation (see 2.1.1 and 2.2).

6.3 International standardization agreements.

This section is not applicable to this standard.

6.3.1 International Standardization Agreements (STANAGs). This section is not applicable to this standard.

6.3.2 Quadripartite Standardization Agreements (QSTAGs). This section is not applicable to this standard.

6.3.3 Air Standardization Coordinating Committee Agreements (ASCC). This section is not applicable to this standard.

6.3.4 International MC&G Agreements. This section is not applicable to this standard.

6.3.5 Executive Orders. This section is not applicable to this standard.

6.3.6 InterAgency Agreements. This section is not applicable to this standard.

6.3.7 Other Documentation.

ANSI 3.41-1974
ANSI/IEEE STD 754-1985
IEEE 1003.1
ISO/IEC 10149
ISO/IEC 10777
ISO/IEC DIS 10089
ISO/IEC DIS 11319

6.4 Subject term (key word) listing.

Data interchange formats
Geospatial databases
Image data structures
Map data structures
Mapping, charting, and geodesy
Raster Product Format

APPENDIX

BIT STRING, FIELD AND GROUP NAME CROSS-REFERENCE.

10. GENERAL

10.1 Scope. This appendix lists the page numbers where each bit string name, each elementary field name and each group name may be found in this standard. This information is intended for reference and for ensuring that names that may be assigned in future versions of this standard do not duplicate existing names.

20. APPLICABLE DOCUMENTS

30. BIT STRINGS.

Item name	**Pages**
/biased exponent/	12-14
/compression lookup value/	46, 54, 60
/fraction/	12-14
/image code/	20, 21, 29, 49, 55-59, 65
/sign bit/	12-14
/transparent output pixel code/	48, 64

40. FIELDS.

Item name	**Pages**
<areal coverage sequence number>	44, 50
<attribute id>	44, 50
<attribute offset record length>	49, 51
<attribute offset table offset>	49, 51
<attribute record offset>	50-51
<boundary rectangle record length>	27, 34
<boundary rectangle record number>	33, 35
<boundary rectangle table offset>	34
<color converter offset record length>	47, 51, 67-68
<color converter offset table offset>	47, 52, 67-68
<color converter record length>	47, 52, 67-68
<color converter table id>	48, 52, 67-68

Item name	Pages
<color converter table offset>	48, 52, 57, 67-68, 71
<color/grayscale element>	47, 52, 67-68
<color/grayscale element length>	47, 52, 55, 67-68, 70
<color/grayscale offset record length>	47, 52, 67-68
<color/grayscale table id>	47, 52, 67-68
<color/grayscale table offset>	47, 52, 67-68
<colormap offset table offset>	47, 52, 68, 70
<colortable index record length>	32, 36
<colortable index table offset>	32, 36
<comment>	35
<component aggregate length>	30, 34, 36, 43, 45, 53, 66-67, 70
<component id>	27, 30, 34, 36, 43, 45, 53, 66-67, 70
<component length>	30, 34, 37, 43-45 53, 66-68, 70
<component location>	34, 37, 43, 44-45, 53, 68, 70
<component location record length>	34, 37, 44-45, 53, 67, 70
<component location table offset>	34, 37, 44-45, 53, 67, 70
<compression algorithm id>	46, 53, 55
<compression lookup offset table offset>	46, 53
<compression lookup table id>	46, 53
<compression lookup table offset>	46, 54
<compression lookup table offset record length>	46, 54
<compression lookup value bit length>	46, 54
<compression parameter id>	46, 51, 54
<compression parameter offset record length>	46, 54
<compression parameter offset table offset>	46, 53
<compression parameter record offset>	46, 53
<compression parameter value>	46, 54

Item name	**Pages**
<compression ratio>	34, 37
<corner coordinates record length>	49, 55
<data series and zone>	17-18
<dummy file name>	68, 70
<east-west/horizontal resolution>	46, 55
<explicit areal coverage record length>	34, 55
<explicit areal coverage table offset>	49, 55
<extension>	16-17
<external color/grayscale file name>	36-37, 47, 56
<file name>	34, 37, 45, 55, 68, 70
<frame file index record length>	35, 37
<frame file index table offset>	35, 37
<frame file name>	35, 38
<frame file security classification>	35, 38
<frame file security country/international code>	35, 38
<frame file security release marking>	35, 38
<frame location column number>	35, 38
<frame location row number>	35, 38
<geographic location>	19, 35, 38
<governing specification date>	45, 55, 68, 71
<governing specification number>	45, 55, 68, 71
<governing standard date>	34, 38
<governing standard number>	34, 39
<header section length>	34, 39, 45, 56, 68, 71
<highest security classification>	35, 39
<histogram element>	47, 56, 69, 71
<histogram record length>	47, 56, 69, 71
<histogram table offset>	47, 56, 69, 71
<image code bit length>	49, 56
<lat>	50, 56, 59
<latitude/vertical interval>	35, 39, 45, 57

Item name	**Pages**
<little/big endian indicator>	34, 39, 45, 57, 68, 71
<location section length>	34, 39, 44, 57, 68, 71
<location section location>	26, 30, 34, 39, 42, 45, 57, 65, 68, 71
<long>	49, 57, 59
<longitude/horizontal interval>	35, 39, 45, 57
<new file name>	50, 57, 60-61
<new/replacement/update indicator>	34, 39, 45, 57, 68, 71
<north-south/vertical resolution>	35, 39, 45, 57
<northeast/upper right latitude>	35, 40, 45, 58
<northeast/upper right longitude>	35, 40, 45, 58
<northwest/upper left latitude>	35, 40, 45, 58
<northwest/upper left longitude>	35, 40, 45, 58
<number of attribute offset records>	49, 58
<number of boundary rectangle records>	34, 40
<number of color converter offset records>	47, 58, 68, 72
<number of color converter records>	48, 58, 69, 72
<number of color/grayscale offset records>	47, 58, 68, 72
<number of color/grayscale records>	47, 58, 63, 68, 72
<number of colortable index records>	36, 40
<number of component location records>	34, 40, 45, 58, 68, 72
<number of compression lookup offset records>	46, 59
<number of compression lookup records>	46, 59
<number of compression parameter offset records>	46, 59
<number of explicit areal coverage records>	49, 59
<number of frame file index records>	35, 40
<number of frames in east-west or left-right direction>	35, 38, 40
<number of frames in north-south or up-down direction>	35, 38, 40

Item name	**Pages**
<number of image codes per row>	49, 59
<number of image rows>	49, 59
<number of output columns per subframe>	48, 59
<number of output rows per subframe>	48, 59
<number of pathname records>	35, 41
<number of related image description records>	50, 59
<number of replace/update records>	51, 59
<number of spectral band lines per image row>	48, 59
<number of spectral band tables>	48, 60
<number of spectral groups>	48, 60
<number of subframe tables>	48, 60
<number of subframes in east-west or left-right direction>	48, 60, 62-63
<number of subframes in north-south or up-down direction>	48, 60, 62-63
<number of values per compression lookup record>	46, 60
<number of vertices>	50, 60
<old file name>	51, 57, 60-61
<parameter id>	50, 51, 60-61
<parameter offset>	43, 44, 60
<parameter value>	43, 44, 49, 51, 61
<pathname>	31, 35, 39, 41
<pathname length>	35, 41
<pathname record offset>	35, 41
<period>	16-17
<producer>	34, 41
<product data type>	34, 41
<reference designator>	16-18, 24-26
<related image description record length>	50, 61
<related image description table offset>	50, 61

Item name	**Pages**
<related image file name>	50, 61
<related image pathname>	51, 61
<related image pathname length>	51, 61
<related image pathname offset>	50, 61
<relationship code>	50-51, 62
<replace/update record length>	51, 62
<replace/update status>	26, 51, 62
<replace/update table offset>	51, 62
<scale or resolution>	34, 41
<security classification>	27, 34, 38, 41, 45, 62, 67, 71
<security country/international code>	34, 41, 45, 62, 68, 72
<security release marking>	34, 41, 45, 62, 68, 72
<source color/grayscale table offset>	48, 58, 63, 69, 72
<southeast/lower right latitude>	35, 42, 45, 63
<southeast/lower right longitude>	35, 42, 45, 63
<southwest/lower left latitude>	35, 42, 45, 63
<southwest/lower left longitude>	35, 42, 45, 63
<subframe mask table offset>	48, 63
<subframe offset>	48, 63
<subframe sequence record length>	48, 64
<target color/grayscale table entry number>	48, 64, 69, 72-73
<target color/grayscale offset table offset>	48, 63, 64, 69, 73
<transparency mask table offset>	48, 64
<transparency sequence number>	49, 64
<transparency sequence record length>	48, 64
<transparent output pixel code length>	49, 65
<zone>	34, 42

50. GROUPS.

Item name	Pages
[attribute offset record]	44, 49, 51, 58
[attribute offset table]	44, 49, 51
[attribute record]	50-51
[attribute section subheader]	49
[attribute section]	29, 42-43, 49, 58
[attribute subsection]	44, 49, 51
[attribute table]	43, 50
[boundary rectangle record]	27, 34, 38-42
[boundary rectangle section subheader]	33
[boundary rectangle section]	20, 30-31, 34, 40
[boundary rectangle subsection]	33-34
[boundary rectangle table]	33-34
[color converter offset record]	47, 51-52, 58, 63-64, 66, 69, 73
[color converter offset table]	47, 58, 67, 73-74
[color converter record]	47, 52, 58, 64, 67, 69, 73-74
[color converter subsection]	47, 52, 67, 69
[color converter table]	43, 48, 52, 58, 64, 66, 67, 69, 73-74
[color/grayscale element group]	47, 69
[color/grayscale offset record]	47, 52, 56, 58, 63, 67-68, 70, 73
[color/grayscale offset table]	63-64, 72-73
[color/grayscale record]	47, 58, 69, 72
[color/grayscale section]	42-43, 47, 65-66, 68, 70
[color/grayscale section subheader]	47, 68
[color/grayscale table]	43, 47, 52, 58, 64, 66, 67, 69, 72-73
[colormap offset table]	47, 52-53, 58, 68, 70, 72

Item name	Pages
[colormap subsection]	47, 52-53, 56, 63-64, 68, 70-72, 74
[colortable file index section]	30, 33
[colortable index record]	36, 40
[colortable index section]	36, 40
[colortable index section subheader]	36
[colortable index subsection]	36
[colortable index table]	36
[component location record]	34, 36-37, 40, 43-45, 53, 58, 66, 68, 70, 72
[component location table]	34, 37, 45, 53, 68, 70
[compression lookup offset record]	46
[compression lookup offset table]	46, 53
[compression lookup record]	46, 53
[compression lookup subsection]	46, 54
[compression lookup table]	46, 54
[compression lookup table offset record]	52, 54
[compression parameter offset record]	46, 54
[compression parameter offset table]	46, 54
[compression parameter record]	46, 54
[compression parameter subsection]	36, 46, 53, 55
[compression section]	36, 42-43, 45, 53, 70
[compression section subheader]	45
[corner coordinates record]	50, 55
[coverage section]	19, 29, 42, 43, 45
[coverage section subheader]	45
[explicit areal coverage record]	50, 55
[explicit areal coverage subsection]	50, 55
[explicit areal coverage table]	50, 55, 59
[external color/grayscale file]	15-17, 30, 33, 36-37, 65-71

Item name	Pages
[file name]	16-17, 24
[frame directory]	15-16
[frame file]	15-26, 29-31, 33, 37-65, 71-72
[frame file index record]	19, 31, 33, 35, 38, 40-41
[frame file index section]	30-31, 35, 39-40
[frame file index section subheader]	35
[frame file index subsection]	35, 35, 41
[frame file index table]	31, 35, 37
[header section]	16-17, 26, 30, 34, 39, 42, 45, 56, 65-66, 68
[histogram element group]	47, 56, 69, 70
[histogram record]	47, 56, 69, 71
[histogram subsection]	66
[histogram table]	47, 56, 66, 69, 71
[image description subheader]	48
[image display parameters subheader]	49
[image row]	29, 49, 56, 59
[image section]	42-43, 48, 55, 59-60, 70
[location section]	22, 26, 27, 30, 34, 36-37, 39-40, 42-43, 45, 53, 57-58, 65-66, 68, 70-72
[lookup table]	53
[lookup table directory]	15-16
[mask subheader]	48
[mask subsection]	23, 48, 63-64

Item name	**Pages**
[nitf message]	66, 70-71
[pathname record]	31, 35, 41
[pathname table]	31, 35, 41
[related image description record]	50, 59, 61-62
[related image description table]	50, 59, 61
[related image pathname record]	51, 61
[related image pathname table]	50, 61
[related images section]	42, 44, 50
[related images section subheader]	50
[related images subsection]	50, 61
[replace/update record]	24-26, 51, 57, 59, 60-62
[replace/update section]	24, 42, 44, 51, 60
[replace/update section subheader]	51
[replace/update subsection]	51, 62
[replace/update table]	44, 51, 62
[row record]	56
[row vector]	56
[rpf root directory]	15116
[spatial data subsection]	22, 28, 49, 63
[spectral band line]	29, 49
[spectral band table]	29, 49, 56
[spectral group]	28, 29, 49, 56
[subframe mask offset table]	62
[subframe mask row]	48
[subframe mask spectral group]	48
[subframe mask subsection]	63
[subframe mask table]	21, 22, 23, 48, 63
[subframe sequence record]	48, 63
[subframe subsection]	55
[subframe table]	29, 49, 59-60, 63-64

Item name	**Pages**
[subframe table within each [spectral group]	29
[subordinate directory]	15
[subordinate frame directory]	16
[table of contents file]	15-16, 19-20, 27, 30, 33-39
[transparency mask offset table]	63
[transparency mask row]	49
[transparency mask subsection]	64
[transparency mask table]	23, 49
[transparency sequence record]	49, 64
[transparency mask table]	22-23

INDEX

SUBJECT	SECTION	PAGE
Accuracy	4.1	6
ADRG	2.1.1, 4.a.(1)	2, 6
ADRI	2.1.1, 4.a.(2)	2, 6
American National Standard Code for Information Interchange	see ASCII	
ANSI 3.41-1974	2.2, 4.4.2.4	3, 14
ANSI/IEEE Std. 754-1985	2.2, 4.4.2.3	3, 12
ARC Digital Raster Imagery	see ADRI	
ARC Digitized Raster Graphics	see ADRG	
ASCII	2.2, 4.4.2.4	3, 14
Attributes, areal coverage	5.1.12	29
Backward compatibility	4.d., 5.1.8	6, 26-27
Big endian recording	see Recording order, big endian	
Data type	4.3.2, 4.4.2	8-11
Datum	2.1.2	3
Directory naming	4.5.4.1	16
Directory structure	4.3.2, 4.5.3, 4.5.4.1	7, 15-16
DMA TR 8350.2	2.1.2, 4.2	3, 6
DMA TM 8358.1	2.1.2, 5.2.1	3, 38
Ellipsoids	2.1.2	3
File naming	4.5.4	16
File naming, external color/grayscale file	4.5.4.3	16
File naming, frame file	4.5.4.4	17
File naming, table of contents file	4.5.4.2	16
File structure, external color/grayscale file	5.2.3	65-73
File structure, frame file	5.2.2	42-65
File structure, table of contents file	5.2.1	30-42
Frame structure	5.1.2, 5.1.3	19-20

SUBJECT	SECTION	PAGE
Frame, updating and replacing	5.1.6	23-26
Grid Reference Systems	2.1.2	3
Grids	2.1.2	3
Horizontal datum	4.2	6
IEEE 1003.1	2.2, 4.5.2.c	3, 15
Image structure, band interleaved by line format	5.1.11	29
Image structure, band interleaved by pixel format	5.1.11	29
Image structure, band interleaved by subframe format	5.1.11	29
Image structure, band sequential format	5.1.11	29
ISO 9660	4.5.2.a	15
ISO 13346	4.5.2.b	15
ISO/IEC 10149	2.2, 4.5.1.1	3, 14
ISO/IEC 10777	2.2, 4.5.1.3	3, 15
ISO/IEC 13346	2.2	4
ISO/IEC DIS 10089	2.2, 4.5.1.2	4, 15
ISO/IEC DIS 11319	2.2, 4.5.1.1	4, 15
Length, bit string	4.3.2	8
Length, byte field	4.3.2	8
Length, variable-length field	4.3.2	8
Little endian recording	see Recording order, little endian	
Maps, compressed and uncompressed	1, 4., 5.1.1	1, 6, 19
Masks, frame and subframe	5.1.5	21-23
Masks transparent pixel	5.1.5.b.	22-23
MIL-A-89007	2.1.1, 4.(1)	2, 6
MIL-A-89027	2.1.1, 4.(2)	2, 6
MIL-STD-2400	2.1.1, 5.1.9	2, 27

SUBJECT	SECTION	PAGE
MIL-STD-2407	2.1.1, 5.1.9	2, 27
MIL-STD-2411-1	1.1, 2.1.1, 5.1.10	1, 2, 27
MIL-STD-2411-2	2.1.1, 4.e., 5.2.1.b.(2), 5.2.2.a.(2), 5.2.3.b.(2)	2, 6, 30, 43, 66
National Imagery Transmission Format	see NITF	
New editions and updates	4.c	6
NITF	2.1.1, 4.e	2, 6
Non-polar maps	4.(1)	6
Non-Sequential Recording	2.2, 4.5.2.b	4, 15
Notation	4.3	6
Notation, <x>	4.3.2	7
Notation, /x/	4.3.2	7
Notation, [x]	4.3.2	7
Notation, {x}	4.3.2	7
Notation, "::="	4.3.3	10
Notation, bit field	4.3.2	7
Notation, byte field	4.3.2	7
Notation, decimal	4.3.1	6
Notation, file structure entry	4.3.2	8
Notation, group	4.3.2	7
Notation, ordered sequence	4.3.2	7
Notation, subordination levels	4.3.2	7
Notation, unordered sequence	4.3.2	7
Online use of RPF data	4.b.	6
Optical media cartridge	2.2	4
Packaging standards	5.1.9	27
Polar maps	4.(1)	6
Portable Operating System Interface for Computer Environments (POSIX)	2.2	3
Process, reversible	4.b	6

SUBJECT	SECTION	PAGE
Product specification	1.1. 4.1	1, 6
Raster Product Format	see RPF	
Recording format, ASCII character string	4.4.2.4	14
Recording format, bit string	4.4.2.5	14
Recording format, Boolean field	4.4.2.7	14
Recording format, byte string	4.4.2.6	14
Recording format, floating-point number	4.4.2.3	12
Recording format, real value	4.4.2.3	12
Recording format, signed integer	4.4.2.1	11
Recording format, unsigned integer	4.4.2.2	11
Recording media, 4mm tape	2.2, 4.5.1.4	3, 15
Recording media, 8mm tape	2.2, 4.5.1.3	4, 15
Recording media, CD-Recordable	2.2, 4.5.1.5	4, 15
Recording media, CD-ROM	2.2, 4.5.1.1	3, 14
Recording media, erasable optical disk	4.5.1.2	15
Recording order, big endian	4.4.1.a, 4.4.1.1	10-11
Recording order, little endian	4.4.1.a 4.4.1.1	10
Recording order, non-numeric data	4.4.1.b	10
Recording standards	4.5	14
Reference designator, generating	4.5.4.3.c, 4.5.4.4.c, 5.1.7	16-17, 26
Registered data values	2.1.1, 5.1.10	2, 27
Repeating groups	4.3.2	8-10
RPF	1.1	1
RPF data, online use	4.b	6
Spectral bands	5.1.11	28-29
Subframe structure	5.1.2, 5.1.3	19-20
Text Product Standard	see TPS	
Third-party receivers	4.5	14

SUBJECT	SECTION	PAGE
TPS	2.1.1, 5.1.9	2, 27
Transparent pixels	5.1.4, 5.1.5.b	21-23
Vector Product Format	See VPF	
Vertical datum	4.2	6
Volume and file structure	4.5.2.b	15
VPF	2.1.1, 5.1.9	2, 27
WGS-84	4.2	6

CONCLUDING MATERIAL

Custodian:
DMA - MP

Review activities:
Air Force - 09
Army - PO
Navy - NO, MC

Preparing activity:
DMA - MP

Agent: AFMC

(Project MCGT-00119)

STANDARDIZATION DOCUMENT IMPROVEMENT PROPOSAL

INSTRUCTIONS

1. The preparing activity must complete blocks 1, 2, 3, and 8. In block 1, both the document number and revision letter should be given.
2. The submitter of this form must complete blocks 4, 5, 6, and 7.
3. The preparing activity must provide a reply within 30 days from receipt of the form.

NOTE: This form may not be used to request copies of documents, nor to request waivers, or clarification of requirements on current contracts. Comments submitted on this form do not constitute or imply authorization to waive any portion of the referenced document(s) or to amend contractual requirements.

I RECOMMEND A CHANGE:	1. DOCUMENT NUMBER MIL-STD-2411	2. DOCUMENT DATE *(YYMMDD)* 6 October 1994

3. DOCUMENT TITLE Military Standard for Raster Product Format

4. NATURE OF CHANGE *(Identify paragraph number and include proposed rewrite, if possible. Attach extra sheets as needed)*

5. REASON FOR RECOMMENDATION

6. SUBMITTER

a. NAME *(Last, First, Middle Initial)*	b. ORGANIZATION	
c. ADDRESS *(Include Zip Code)*	d. TELEPHONE *(Include Area Code)* (1) Commercial (2) AUTOVON *(If applicable)*	7. DATE SUBMITTED *(YYMMDD)*

8. PREPARING ACTIVITY

a. NAME	b. TELEPHONE *(Include Area Code)* (1) Commercial	(2) AUTOVON
Defense Mapping Agency ATTN: TI, ST A-10	(703) 285-9238	356-9238
ADDRESS *(Include Zip Code)* 8613 Lee Highway Fairfax, VA 22031-2137	**IF YOU DO NOT RECEIVE A REPLY WITHIN 45 DAYS, CONTACT:** Defense Quality and Standardization Office 5203 Leesburg Pike, Suite 1403, Falls Church, VA. 22041-3466 Telephone (703) 756-2340 AUTOVON 289-2340	