NOT MEASUREMENT SENSITIVE

MIL-STD-2301A
5 June 1998
SUPERSEDING
ON 1 OCTOBER 1998
MIL-STD-2301
18 June 1993

DEPARTMENT OF DEFENSE
INTERFACE STANDARD

COMPUTER GRAPHICS METAFILE (CGM) IMPLEMENTATION STANDARD FOR THE NATIONAL IMAGERY TRANSMISSION FORMAT STANDARD

AMSC N/A **AREA IPSC**

FOREWORD

1. This standard is approved for use by all Departments and Agencies of the Department of Defense (DOD).

2. The National Imagery Transmission Format Standard (NITFS) is the standard for formatting digital imagery and imagery-related products and exchanging them among the DOD, other members of the Intelligence Community (IC), as defined by Executive Order 12333, and other departments and agencies of the United States Government

3. The NITFS Technical Board (NTB) developed this standard based upon currently available technical information.

4. The DOD and other IC members are committed to interoperability of systems used for formatting, transmitting, receiving, and processing imagery and imagery-related information. This standard describes a Computer Graphics Metafile (CGM) implementation and establishes its application within the NITFS.

5. As depicted on the cover, MIL-STD-2301A will supersede MIL-STD-2301, Computer Graphics Metafile (CGM) Implementation Standard for the National Imagery Transmission Format Standard, 18 June 1993, on 1 October 1998. The 1 October 1998 supersession date coincides with the date on which MIL-STD-2500B, National Imagery Transmission Format Version 2.1 for the National Imagery Transmission Format Standard, 22 August 1997 will supersede MIL-STD-2500A, National Imagery Transmission Format (Version 2.0) for the National Imagery Transmission Format Standard, 12 October 1994.

6. Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be of use in improving this document should be addressed to National Imagery and Mapping Agency, NIMA/SES, 12310 Sunrise Valley Drive, Reston, VA 20191-3449, Attention: NTB Chair, by using the Standardization Document Improvement Proposal (DD Form 1426) appearing at the end of this document or by letter.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

CONTENTS

1. SCOPE

1.1 Scope. This standard establishes the requirements necessary to implement Computer Graphics Metafiles (CGMs) used for the representation of symbol graphics in the National Imagery Transmission Format Standard (NITFS).

1.2 Purpose. This standard provides technical detail of CGM commands, formats, and implementation used for the CGM implementation for NITFS.

1.3 Applicability. This standard is applicable to the Department of Defense (DOD), other Intelligence Community (IC) members, and other US Government departments and agencies. This standard defines the subset of commands (correlated with the minimum implementation subset of commands specified in MIL-D-28003) applicable for graphic annotation of imagery within the NITFS. It is mandatory for all Secondary Imagery Dissemination Systems (SIDS) in accordance with the memorandum by the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, ASD(C^3I), Subject: National Imagery Transmission Format Standard (NITFS), 12 August 1991. This directive will be implemented in accordance with the National Imagery and Mapping Agency (NIMA) N0105-98, National Imagery Transmission Format Standard (NITFS) Standards Compliance and Interoperability Test and Evaluation Program Plan (supersedes Joint Interoperability and Engineering Organization (JIEO) Circular 9008), and NIMA NNPP 1.0, The National Imagery Transmission Format Standard Five Year Program Plan. New equipment and systems, those undergoing major modification, or those capable of rehabilitation will conform to this standard.

2. APPLICABLE DOCUMENTS

2.1 General. The documents listed in this section are specified in sections 3, 4, and 5 of this standard. This section does not include documents cited in other sections of this standard or recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements documents cited in sections 3, 4, and 5 of this standard, whether or not they are listed.

2.2 Government documents.

2.2.1 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 issue of these documents are those listed in the issue of the Department of Defense Index of Specifications and Standards (DODISS) and supplement thereto, cited in this solicitation.

STANDARDS

FEDERAL

FED-STD-1037B - Telecommunications: Glossary of Telecommunication Terms, 3 June 1991.

FEDERAL INFORMATION PROCESSING STANDARDS (FIPS)

FIPS PUB 128-2 - Computer Graphics Metafile (CGM) [adaptation of American National Standards Institute/International Organization for Standardization (ANSI/ISO) 8632.1-4:1992[1994].

MILITARY

MIL-STD-2500A - National Imagery Transmission Format (Version 2.0) for the National Imagery Transmission Format Standard 12 October 1994.

MIL-STD-2500B	-	National Imagery Transmission Format Version 2.1 for the National Imagery Transmission Format Standard, 22 August 1997.
MIL-D-28003A	-	Military Representation for Communication of Illustration Data: CGM Application Profile, 15 November 1991.

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

(Copies of Federal Information Processing Standards (FIPS) are available to Department of Defense activities from the Standardization Documents Order Desk, 700 Robbins Avenue, Building 4D, Philadelphia, PA 19111-5094. Others must request copies of FIPS from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161-2171.)

2.2.2 Other Government documents, drawings, and publications. The following other Government documents, drawings, and publications form a part of this document to the extent specified herein. Unless otherwise specified, the issues are those cited in the solicitation.

NATIONAL IMAGERY AND MAPPING AGENCY PUBLICATIONS

NIMA N105-97	-	National Imagery Transmission Format Standard (NITFS) Standards Compliance and Interoperability Test and Evaluation Program Plan (supersedes JIEO Circular 9008).
NIMA NNPP-97	-	The National Imagery Transmission Format Standard Program Plan.

(Copies of NIMA documents can be obtained from the web at http://www.nima.mil.)

2.3 Non-Government publications. The following document(s) form 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.

AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI)

ANSI X3.4 - 1986	-	American National Standard Code for Information Interchange (ASCII), 1986.

(Application for copies should be addressed to the American National Standards Institute, 1430 Broadway, New York, NY 10018-3308.)

2.4 Order of precedence. In the event of a conflict between the text of this document and the references cited herein, 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

3.1 Acronyms used in this standard. The acronyms used in this standard are defined as follows.

a.	ANSI	American National Standards Institute
b.	ASCII	American Standard Code for Information Interchange
c.	ASD(C^3I)	Assistant Secretary of Defense for Command, Control, Communications and Intelligence

d.	CGM	Computer Graphics Metafile
e.	DISA	Defense Information Systems Agency
f.	DOD	Department of Defense
g.	DODISS	Department of Defense Index of Specifications and Standards
h.	FIPS	Federal Information Processing Standard
i.	IC	Intelligence Community
j.	ISO	International Organization for Standardization
k.	JIEO	Joint Interoperability and Engineering Organization
l.	LSB	Least Significant Bit
m.	MOA	Memoranda of Agreement
n.	MSB	Most Significant Bit
o.	NIMA	National Imagery and Mapping Agency
p.	NITF	National Imagery Transmission Format
q.	NITFS	National Imagery Transmission Format Standard
r.	NTB	National Imagery Transmission Format Standard Technical Board
s.	RGB	Red, Green, Blue
t.	SIDS	Secondary Imagery Dissemination System
u.	VDC	Virtual Device Coordinates

3.2 Character. 1. A letter, digit, or other symbol that is used as part of the organization, control, or representation of data. 2. One of the units of an alphabet. Note: For MIL-STD-2301A, a character (UT-1 ISO 4873, 1-octet coded Universal Multiple-Octet Coded Character Set (UCS), Basic Latin and Latin-1 Supplement) is constraint to Basic Latin (0020-007E) and Latin-1 Supplement (00A0-00FF). The unsigned integer value of 0 (null) is also an allowed character value.

3.3 Commands. For MIL-STD-2301A, commands are CGM statements that denote a state to act upon when CGM is read sequentially. The words "command" and "element" are used synonymously throughout MIL-STD-2301A.

3.4 Computer Graphics Metafile (CGM). CGM is a set of basic elements for a computer graphics data interface usable by many graphics-producing systems and applications.

3.5 Elements. For MIL-STD-2301A, elements are CGM statements that denote a state to act upon when the CGM is sequentially read. The words "command" and "element" are synonymously used in MIL-STD-2301A.

3.6 Integer parameters. For MIL-STD-2301A, all integer parameters are 16-bit two's complement signed integers except where specified. Each 16-bit word is numbered from most significant bit to least significant bit using 15 to zero (as illustrated on figure 1). When a 16-bit two's complement integer is used as a parameter in a CGM metafile, the high order byte of the integer is represented as the 8 most significant bits. That is, bits 15 through 8. Bits 7 through zero represent the low order byte of the integer. Note: This is also known as the "Big-Endian" or "Network Byte Order" representation for 16-bit integers.

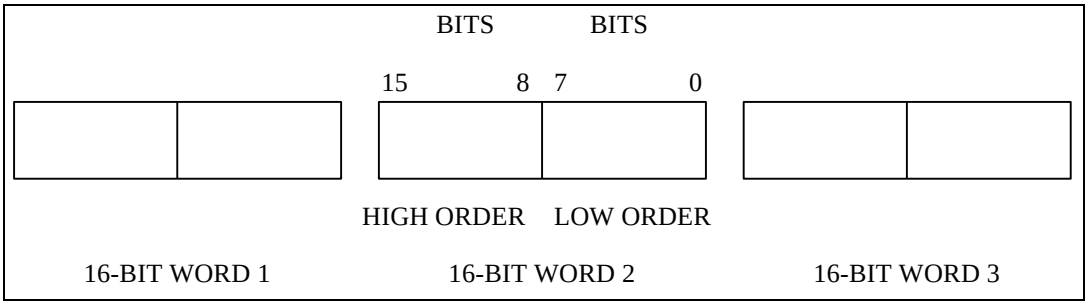

FIGURE 1. Integer parameters.

3.7 Virtual Device Coordinates (VDC) space. The VDC space defines a coordinate system that is overlaid onto an image to which CGM elements are referenced.

4. GENERAL REQUIREMENTS

4.1 CGM commands. The CGM implementation for NITFS is a subset of the Federal Information Processing Standard FIPS PUB 128 and complies with the specifications established in FIPS PUB 128. The CGM structure is explained fully in the FIPS PUB 128 document. The following describes the required CGM commands that the CGM implementation for NITFS must support for input interpretation and output generation and are grouped by element class. All CGM commands used in this subset are listed by element class along with their encoding in the CGM Interface Input Requirements and CGM Interface Output Requirements sections of this document. When CGM commands are encountered that are beyond the minimally compliant CGM implementation listed below, the interpreter is responsible for interpreting, discarding, or substituting for these commands.

a. Metafile Delimiter Elements
   - BEGIN METAFILE
   - BEGIN PICTURE
   - BEGIN PICTURE BODY
   - END PICTURE
   - END METAFILE

b. Metafile Descriptor Elements
   - METAFILE VERSION (version = 1)
   - METAFILE ELEMENT LIST
   - METAFILE DESCRIPTION
   - FONT LIST

c. Metafile Picture Descriptor Elements
   - COLOR SELECTION MODE (1 = direct)
   - EDGE WIDTH SPECIFICATION MODE (0 = absolute)
   - LINE WIDTH SPECIFICATION MODE (0 = absolute)
   - VDC EXTENT

d. Metafile Graphical Primitives with Associated Attributes
   Text Primitive Element with Attributes
      TEXT COLOR
      CHARACTER HEIGHT
      TEXT FONT INDEX
      CHARACTER ORIENTATION
      TEXT
   Filled-Area Primitive Elements with Attributes
      FILL COLOR
      INTERIOR STYLE (1= solid, 3=hatch, or 4 = empty)
      EDGE VISIBILITY (0=off, 1 = on)
      EDGE WIDTH
      EDGE TYPE (1=solid, 2=dashed, 3=dot, 4=dash-dot, 5=dash-dot-dot)
      EDGE COLOR
      POLYGON
      POLYGON SET
      ELLIPSE
      ELLIPTICAL ARC CLOSE
      RECTANGLE
      CIRCLE
      CIRCULAR ARC CENTER CLOSE
   Line Primitive Elements with Attributes
      LINE WIDTH
      LINE TYPE (1=solid, 2=dashed, 3=dot, 4=dash-dot, 5=dash-dot-dot)
      LINE COLOR
      POLYLINE
      ELLIPTICAL ARC
      CIRCULAR ARC CENTER

e. Metafile Control Elements
      TRANSPARENCY
      AUXILIARY COLOR

4.2 CGM binary encoding. CGM commands in the CGM implementation for NITFS are encoded using the binary encoding method described in the FIPS PUB 128. Metafile elements will be represented in the binary encoding in either short-form commands or long-form commands. For short-form and long-form commands, the 4 most significant bits (MSB) identify the element class in which the command belongs (for example, Delimiter Elements, Descriptor Elements) and the fifth through eleventh bits identify the element ID (for example, BEGIN METAFILE, END METAFILE). For the short-form command, the five least significant bits (LSB) specify the length, in bytes, of the parameter list. This form is used to specify parameter lists that are less than 31 LONG. For the long-form command, the 5 LSB are set to the binary value "11111" (decimal 31). In this case, the next 2 bytes are interpreted as a signed positive integer containing the length of the parameter list. For short-form and long-form commands, the first byte of a string parameter specifies the length of the string. If necessary, parameters are padded with a trailing null byte (0) to ensure that all subsequent commands begin on a 16-bit word boundary. Note, the trailing null byte is not included in the parameter list length. In the figures contained in this standard, all numbers are decimal unless preceded by "0x" indicating hexadecimal notation.

TABLE I. Short form of CGM command.

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
element class				element id							parameter list length				
Parameters															
⋮															

TABLE II. Long form of CGM command.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
element class				element id							31				
parameter list length															
Parameters															
⋮															

4.3 CGM element flow. The following sequence of commands is used to describe all CGM graphics required for the CGM implementation for NITFS. The following flow diagram displays all the CGM commands for the NITFS implementation that could be contained in a metafile. The commands enclosed in brackets are optional within a given metafile and are used only when needed to describe the desired graphical symbol. Commands following the BEGIN PICTURE BODY command may be repeated to describe multiple graphics within the same metafile. The commands are executed in sequential order.

BEGIN METAFILE

METAFILE VERSION
METAFILE ELEMENT LIST
METAFILE DESCRIPTION
[FONT LIST]

BEGIN PICTURE

COLOR SELECTION MODE
[EDGE WIDTH SPECIFICATION MODE]
[LINE WIDTH SPECIFICATION MODE]
VDC EXTENT

BEGIN PICTURE BODY

[TRANSPARANCY]
[AUXILLIARY COLOR]
[TEXT COLOR]
[CHARACTER HEIGHT]
[TEXT FONT INDEX]
[CHARACTER ORIENTATION]
[TEXT]
[FILL COLOR]
[INTERIOR STYLE]
[EDGE VISIBILITY]
[EDGE WIDTH]
[EDGE TYPE]
[EDGE COLOR]
[POLYGON]
[POLYGON SET]
[ELLIPSE]
[ELLIPTICAL ARC CLOSE]
[RECTANGLE]
[CIRCLE]
[CIRCULAR ARC CENTER CLOSE]

[LINE WIDTH]
[LINE TYPE]
[LINE COLOR]
[POLYLINE]
[ELLIPTICAL ARC]

[CIRCULAR ARC CENTER]

END PICTURE

END METAFILE

Default elements as identified in TABLE CXVIII CGM element defaults for output, may optionally be included in a CGM file. If the default element(s) are included, they can only contain those values as identified in tables CXVIII.

5. DETAILED REQUIREMENTS

5.1 Interface requirements. The following subsections list the required CGM commands along with the binary encoding method as described in the FIPS PUB 128 document. The words "command" and "element" are used synonymously throughout this document.

5.1.1 CGM interface input requirements.

5.1.1.1 Delimiter elements. The Delimiter elements define boundaries for significant structures within the metafile.

5.1.1.1.1 Begin Metafile element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Begin Metafile element using the following formats. The Begin Metafile element name is represented using the character string C1, C2, ... Cn with length n.

TABLE III. Begin Metafile padded, short form input.

MSB 15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0 LSB
0	1	parameter list length
length = n (even)		C1
C2		⋮
Cn		0

TABLE IV. Begin Metafile nonpadded, short form input.

MSB 15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0 LSB
0	1	parameter list length
length = n (odd)		C1
C2		⋮
C(n-1)		Cn

TABLE V. Begin Metafile padded, long form input.

MSB 15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0 LSB	
0	1	31	= 0x003F
parameter list length			
length = n (even)		C1	
C2		⋮	
Cn		0	

TABLE VI. Begin Metafile nonpadded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
0	1	31	= 0x003F
parameter list length			
length = n (odd)	C1		
C2	:		
C(n-1)	Cn		

5.1.1.1.2 Begin Picture element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Begin Picture element using the following formats. The Begin Picture element name is represented using the character string C1, C2, ... Cn with length n.

TABLE VII. Begin Picture padded, short form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0
0	3	parameter list length
length = n (even)	C1	
C2	:	
Cn	0	

TABLE VIII. Begin Picture nonpadded, short form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0
0	3	parameter list length
length = n (odd)	C1	
C2	:	
C(n-1)	Cn	

TABLE IX. Begin Picture padded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
0	3	31	= 0x007F
parameter list length			
length = n (even)	C1		
C2	:		
Cn	0		

TABLE X. Begin Picture nonpadded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
0	3	31	= 0x007F
parameter list length			
length = n (odd)	C1		
C2	:		
C(n-1)	Cn		

5.1.1.1.3 Begin Picture Body element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Begin Picture Body element using the following format.

TABLE XI. Begin Picture Body input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				4							0					= 0x0080

5.1.1.1.4 End Picture element input. The CGM implementation for NITFS shall provide the capability to input and interpret the End Picture element using the following format.

TABLE XII. End Picture input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				5							0					= 0x00A0

5.1.1.1.5 End Metafile element input. The CGM implementation for NITFS shall provide the capability to input and interpret the End Metafile element using the following format.

TABLE XIII. End Metafile input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				2							0					= 0x0040

5.1.1.2 Metafile Descriptor elements. The Metafile Descriptor elements describe the functional content, default conditions, and characteristics of the Metafile.

5.1.1.2.1 Metafile Version element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Metafile Version element (version 1) using the following format.

TABLE XIV. Metafile Version input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				1							2					= 0x1022
1																= 0x0001

5.1.1.2.2 Metafile Description element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Metafile Description element using the following formats. The Metafile Description element name is represented using the character string C1, C2, ... Cn with length n.

TABLE XV. Metafile Description padded, short form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				2							parameter list length				
length = n (even)								C1							
C2								⋮							
Cn								0							

TABLE XVI. Metafile Description nonpadded, short form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				2							parameter list length				
Length = n (odd)								C1							
C2								⋮							
C(n-1)								Cn							

TABLE XVII. Metafile Description padded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
1	2	31	= 0x105F
parameter list length			
length = n (even)	C1		
C2	:		
Cn	0		

TABLE XVIII. Metafile Description nonpadded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
1	2	31	= 0x105F
parameter list length			
length = n (odd)	C1		
C2	:		
C(n-1)	Cn		

5.1.1.2.3 Metafile Element List element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Metafile Element List command using the following format. The metafile elements are chosen from specified CGM command subsets. When the second parameter is one, the metafile element is "Drawing Plus Control Set."

TABLE XIX. Metafile Element List input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
1	2	6	= 0x1166
1			= 0x0001
-1			= 0xFFFF
1			= 0x0001

5.1.1.2.4 Font List element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Font List element using the following formats. The first font name defined in the list is of length x with the font name given as the character string C1, C2, ... Cx assigned to index 1. The last font name defined in the list is of length z with the font name given as the character string C1, C2, ... Cz and assigned to index N. Selection of named fonts is accomplished with the Text Font Index element.

TABLE XX. Font List padded, short form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0
1	13	parameter list length
length = x	C1	
C2	:	
C(x-1)	Cx	
length = z	C2	
C1	:	
Cz	0	

TABLE XXI. Font List nonpadded, short form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0
1	13	parameter list length
length = x	C1	
C2	:	
:	C(x-1)	
Cx	Length = z	
C1	C2	
:	Cz	

TABLE XXII. Font List padded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
1	13	31	= 0x11BF
parameter list length			
length = x	C1		
C2	:		
C(x-1)	Cx		
length = z	C1		
C2	:		
Cz	0		

TABLE XXIII. Font List nonpadded, long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
1	13	31	= 0x11BF
parameter list length			
length = n	C1		
C2	:		
:	C(x-1)		
Cx	Length = z		
C1	C2		
:	Cz		

5.1.1.3 Picture Descriptor elements. The Picture Descriptor elements set the interpretation modes of the attribute elements.

5.1.1.3.1 Color Selection Mode element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Color Selection Mode element using the following format.

TABLE XXIV. Color Selection Mode input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
2	2	2	= 0x2042
1			= 0x0001

5.1.1.3.2 Edge Width Specification Mode element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Edge Width Specification Mode element using the following format. Edge width is specified in "absolute mode," which indicates that the width is specified in source pixels. This command is used to support filled-area primitives.

TABLE XXV. Edge Width Specification Mode input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				5							2					= 0x20A2
0																= 0x0000

5.1.1.3.3 Line Width Specification Mode element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Line Width Specification Mode element using the following format. Line width is specified in "absolute mode," which indicates that the width is specified in pixels. This command is used to support the line primitive.

TABLE XXVI. Line Width Specification Mode input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				3							2					= 0x2062
0																= 0x0000

5.1.1.3.4 VDC Extent element input. The CGM implementation for NITFS shall provide the capability to input and interpret the VDC Extent element using the following format.

TABLE XXVII. VDC Extent input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				6							8					= 0x20C8
X1																
Y1																
X2																
Y2																

This element (illustrated on figures 2 and 3) defines the rectangular extent and orientation of the VDC space (the direction of the positive X and positive Y axes). The extent and orientation of VDC space is indicated by giving the coordinates of lower left hand and upper right hand corners of the VDC extent. The VDC extent space is a one-to-one map from the VDC extent to the source coordinates. The VDC extent origin (0,0) is mapped onto the source at the row and column location given by the SLOC field in the Symbol Subheader for a given image. The Symbol Subheader and the SLOC field are defined in the MIL-STD-2500B document. Note that X1 and X2 cannot be equal, and Y1 and Y2 cannot be equal. Note that the proper Character Orientation element must be present to ensure that CGM text is right side up for a given VDC Extent element. The VDC extent defines the positive 90-degree angle to be the right angle from the positive X-axis to the positive Y-axis.

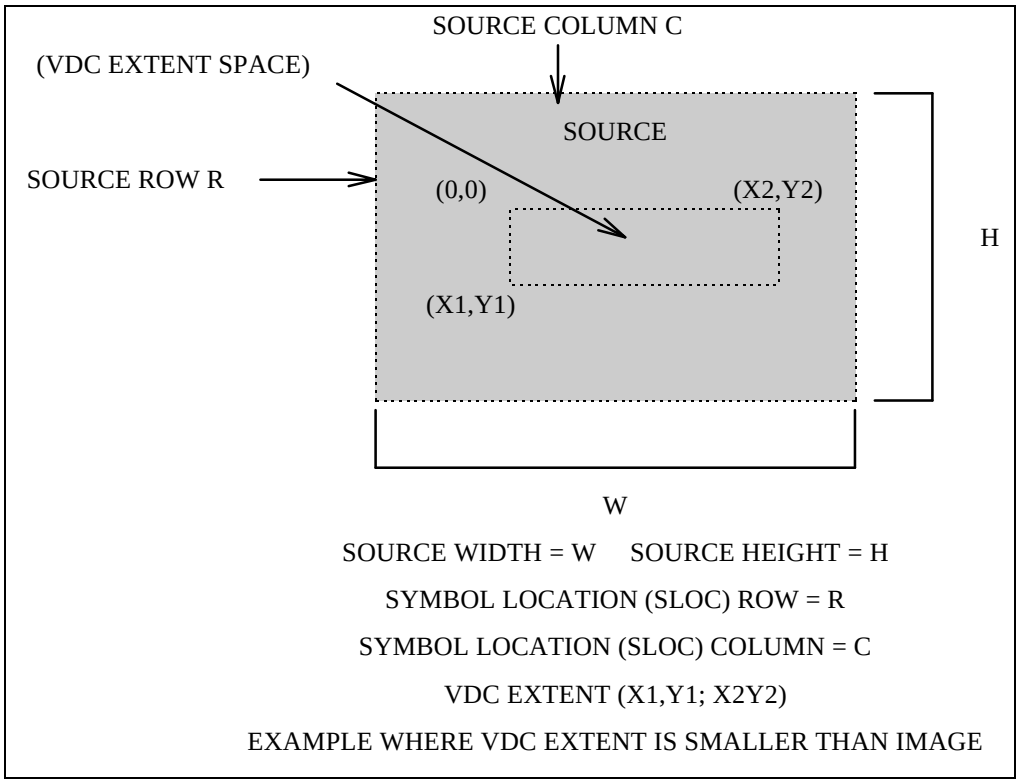

FIGURE 2. Small VDC extent input.

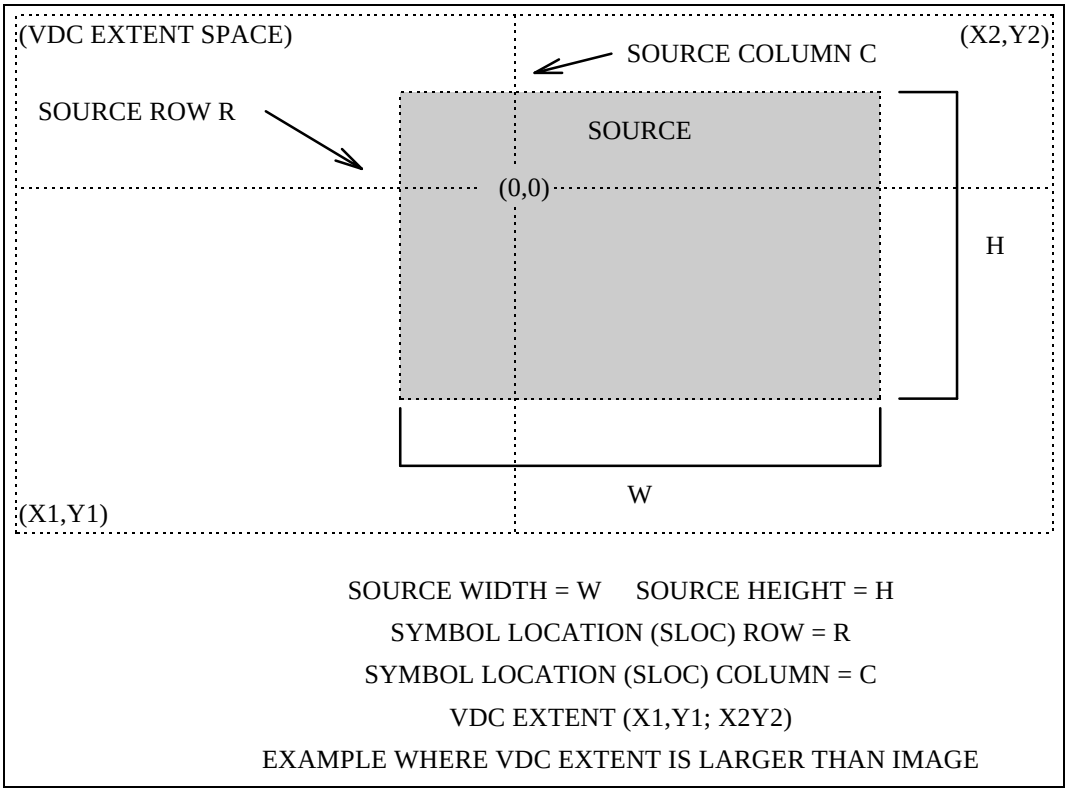

FIGURE 3. Large VDC extent input.

5.1.1.4 Attribute elements. Attribute elements are used to describe the appearance of the Graphical Primitive elements.

5.1.1.4.1 Text Color element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Text Color element using the following format. The Text Color element is used to support the text primitives. Red, Green, and Blue (RGB) values are specified using a single byte, and the last byte of the command shall be a null byte.

TABLE XXVIII. Text Color input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				14							3					= 0x51C3
RED								GREEN								
BLUE								0								

5.1.1.4.2 Character Height element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Character Height element using the following format. Variable character heights, as measured from baseline to capline, shall be supported. The font, if given, is specified by the Font List and Text Font Index elements.

TABLE XXIX. Character Height input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				15							2					= 0x51E2
CHARACTER_HEIGHT																

5.1.1.4.3 Text Font Index element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Text Font Index element using the following format. The Text_Font_Index parameter is the index into the Font List array created from the Font List element for selection of a particular font.

TABLE XXX. Text Font Index input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				10							2					= 0x5142
TEXT_FONT_INDEX																

5.1.1.4.4 Character Orientation element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Character Orientation element using the following format. This command defines the orientation and skew of the character body in subsequent text elements. Since the VDC Extent element is used to establish the direction of the positive X and positive Y axis, the character orientation must be present to ensure that text characters are always upright from left to right when the VDC Extent element defines Y increasing down or X increasing left. The following four cases apply to the Character Orientation element for each possible VDC extent orientation:

a. VDC Extent element with X increasing right and Y increasing up (X1 < X2 and Y1 < Y2).

   Character Orientation element not required or Y=1 and X=1 when Character Orientation present.

b. VDC Extent element with X increasing right and Y increasing down (X1<X2 and Y1>Y2).

   Character Orientation element required with Y=-1 and X=1.

c. VDC Extent element with X increasing left and Y increasing up (X1>X2 and Y1<Y2).

   Character Orientation element required with Y=1 and X=-1.

d. VDC Extent element with X increasing left and Y increasing down (X1>X2 and Y1>Y2).

   Character Orientation element required with Y=-1 and X=-1.

TABLE XXXI. Character Orientation input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
5	16	8	= 0x5208
0			= 0x0000
Y			
X			
0			= 0x0000

5.1.1.4.5 Fill Color element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Fill Color element using the following format. The Fill Color command is used to support the filled-area primitives. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE XXXII. Fill Color input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
5	23	3	= 0x52E3
RED		GREEN	
BLUE		0	

5.1.1.4.6 Interior Style element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Interior Style element using the following format. Filled-area primitives must support the Interior_Style parameter for solid (1), hatch (3) and empty (4).

TABLE XXXIII. Interior Style input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
5	22	2	= 0x52C2
INTERIOR_STYLE			

5.1.1.4.6.1 Hatch Index input. The CGM implementation for NITFS shall provide the capability to input and interpret the Hatch Index element using the following format. The Hatch Index command is used to support the primitive filled-area styles: horizontal (1), vertical (2), positive slope (3), negative slope (4), horizontal/vertical crosshatch (5) and positive/negative slope cross (6).

TABLE XXXIII(A). Hatch Index input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
5	24	2	= 0x5302
HATCH_INDEX			

5.1.1.4.7 Edge Visibility element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Edge Visibility element using the following format. The edge visibility command (0 = off, 1 = on).

TABLE XXXIV. Edge Visibility input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1	LSB 0	
5	30	2	= 0x52C2
EDGE_VISIBILITY			

5.1.1.4.8 Edge Width element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Edge Width element using the following format. Variable edge widths are supported for use in filled-area primitives.

TABLE XXXV. Edge Width input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				28							2					= 0x5382
EDGE_WIDTH																

5.1.1.4.9 Edge Type element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Edge Type element using the following format. The Edge_Type parameter can be solid (1), dashed (2), dotted (3), dash-dot (4), and dash-dot-dot (5).

TABLE XXXVI. Edge Type input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				27							2					= 0x5362
EDGE_WIDTH																

5.1.1.4.10 Edge Color element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Edge Color element using the following format. Edge Color is supported for use in filled-area primitives. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE XXXVII. Edge Color input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				29							3					= 0x53A3
RED								GREEN								
BLUE								0								

5.1.1.4.11 Line Width element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Line Width element using the following format. Variable line widths are supported for use by line primitives.

TABLE XXXVIII. Line Width input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				3							2					= 0x5362
LINE_WIDTH																

5.1.1.4.12 Line Type element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Line Type element using the following format. The Line_Type parameter can be solid (1), dashed (2), dotted (3), dash-dot (4), and dash-dot-dot (5).

TABLE XXXIX. Line Type input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				2							2					= 0x5042
LINE_TYPE																

5.1.1.4.13 Line Color element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Line Color element using the following format. Line colors are supported for use by the line primitive. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE XL. Line Color input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				4							3					= 0x5383
RED								GREEN								
BLUE								0								

5.1.1.5 Graphical Primitive elements. The Graphical Primitive elements describe the visual components of a picture as contained in the CGM.

5.1.1.5.1 Text element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Text element using the following formats. The X and Y parameters give the coordinates of the lower left hand corner of the text string. The flag parameter is an integer (1) indicating that this is the final text in the string. The length parameter is an unsigned byte containing the number of characters in the string. Finally, the text is given as a character string C1, C2, ... Cn with length n. If necessary, the last byte is padded with a zero so that the next command begins on a word boundary.

TABLE XLI. Text padded, short form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							parameter list length					
X																
Y																
1																= 0x0001
length = n(even)								C1								
C2								:								
Cn								0								

TABLE XLII. Text nonpadded, short form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							parameter list length					
X																
Y																
1																= 0x0001
length = n(odd)								C1								
C2								:								
C(n-1)								Cn								

TABLE XLIII. Text padded, long form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							31					= 0x409F
parameter list length																
X																
Y																
1																= 0x0001
length = n(even)								C1								
C2								:								
Cn								0								

TABLE XLIV. Text nonpadded, long form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							31					= 0x409F
parameter list length																
X																
Y																
1																= 0x0001
length = n(odd)								C1								
C2								:								
C(n-1)								Cn								

5.1.1.5.2 Polygon element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Polygon element using the following format. The polygon parameters consist of a list of pairs of coordinates indicating the vertices of a polygon. The first vertex (Vertex1X, Vertex1Y) is connected to the last (Vertex(N)X, Vertex(N)Y) to close the polygon. Polygons are not "clipped" to the image boundary; therefore, some coordinates may specify off-image or off-screen locations, including negative locations. Note, the parameter list length is given as the total number of bytes for all vertex parameters (4N).

TABLE XLV. Polygon short form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
4				7							parameter list length (4N)				
Vertex1X															
Vertex1Y															
Vertex2X															
Vertex2Y															
:															
Vertex(N)X															
Vertex(N)Y															

TABLE XLVI. Polygon long form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				7							31					= 0x40FF
Parameter list length (4N)																
Vertex1X																
Vertex1Y																
Vertex2X																
Vertex2Y																
:																
Vertex(N)X																
Vertex(N)Y																

5.1.1.5.2.1 Polygon Set element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Polygon Set element using the following format. The polygon set parameters consist of a list of pairs of coordinates indicating the vertices of each polygon in the polygon set along with the Edge_Out_Flag indicating the edge visibility and whether the vertex is the last (closure) vertex of the specific polygon in the set. The first vertex of the first polygon (Vertex_11X, Vertex_11Y, Edge_Out_Flag_11) is connected to the last Vertex_1(N)X, Vertex_1(N)Y, Edge_OutFlag_1(N)) vertex in the first polygon followed by the vertices of each successive polygon in like manner. Polygon Sets are not “clipped” to the image boundary; therefore, some coordinates may specify off-image or off-screen locations, including negative locations. The Edge_Out_Flag parameter can be Invisible (0), Visible (1), Close Invisible (2), or Close Visible(2). Note, the parameter list length

is given as the total number of bytes for all vertex parameters (6N).

TABLE XLVII. Polygon short form input.

MSB 15 14 13 12	11 10 9 8 7 6 5	4 3 2 1 0 LSB
4	8	parameter list length (6N)
Vertex_11X		
Vertex_11Y		
Edge_Out_Flag_11		
Vertex_12X		
Vertex_12Y		
Edge_Out_Flag_12		
⋮		
Vertex_(n)(N)X		
Vertex_(n)(N)Y		
Edge_Out_Flag_(n)(N)		

TABLE XLVIII. Polygon Set long form input.

MSB 15 14 13 12	11 10 9 8 7 6 5	4 3 2 1 0 LSB	
4	8	31	= 411F
parameter list length (6N)			
Vertex_11X			
Vertex_11Y			
Edge_Out_Flag_11			
Vertex_12Y			
⋮			
Vertex_1(N)X			
Vertex_1(N)Y			
Edge_Out_Flag_1(N)			
Vertex_21X			
Vertex_21Y			
Edge_Out_Flag_21			
Vertex_22X			
Vertex_22Y			
Edge_Out_Flag_22			
⋮			
Vertex_2(N)X			
Vertex_2(N)Y			
Edge_Out_Flag_2(N)			
⋮			
Vertex_(n)1X			
Vertex_(n)1Y			
Edge_Out_Flag_(n)1			
Vertex_(n)2X			
Vertex_(n)2Y			
Edge_Out_Flag_(n)2			
⋮			
Vertex_(n)(N)X			
Vertex_(n)(N)Y			
Edge_Out_Flag_(n)(N)			

5.1.1.5.3 Ellipse element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Ellipse element using the following format. The ellipse parameters consist of a list of three pairs of coordinates. The first pair, CenterX and CenterY, gives the location of the center of the ellipse. The other two pairs, End1X, End1Y, and End2X, End2Y, specify the end points of the first and second conjugate diameters. The ellipse interior will be as specified by the Interior Style command, and the edge width, type, and color will be as specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE XLIX. Ellipse input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				8							31					= 0x422C
CenterX																
CenterY																
Edn1X																
End1Y																
End2X																
End2Y																

5.1.1.5.4 Polyline element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Polyline element using the following format. The polyline parameters consist of a list of pairs of coordinates indicating the vertices of a polyline (Vertex1X, Vertex1Y) to (Vertex(N)X, Vertex(N)Y). The line width, type, and color are specified by the Line Width, Line Type, and Line Color commands, respectively. Polylines are not "clipped" to the image boundary. Therefore, some coordinates may specify off-image or off-screen locations, including negative locations. Note, the parameter list length is given as the total number of bytes for all vertex parameters (4N).

TABLE L. Polyline short form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
4				1							parameter list length (4N)				
Vertex1X															
Vertex1Y															
Vertex2X															
Vertex2Y															
:															
Vertex(N)X															
Vertex(N)Y															

TABLE LI. Polyline long form input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				1							31					= 0x403F
Parameter list length (4N)																
Vertex1X																
Vertex1Y																
Vertex2X																
Vertex2Y																
:																
Vertex(N)X																
Vertex(N)Y																

5.1.1.5.5 Elliptical Arc element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Elliptical Arc element using the following format. A conjugate diameter pair of an ellipse is a pair of diameters of the ellipse such that a tangent to the ellipse at each endpoint is parallel to the other diameter. The centerpoint (CenterX, CenterY) specifies the center of an ellipse. The conjugate diameter endpoints (End1X, End1Y, and End2X, End2Y) include one endpoint from each conjugate diameter; together with the centerpoint they define the two conjugate diameters of the ellipse. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors respectively. The define arc begins at the intersection of the ellipse and the end ray in the direction defined as follows. A "conjugate radius" is defined to be half of a conjugate diameter. Letting the centerpoint be labeled M, the first conjugate diameter endpoint P_1, and the second conjugate diameter endpoint P_2, then the line segments M-P_1 and M-P_2 define two conjugate radii, referred to in what follows as the first conjugate radius and the second conjugate radius respectively. The conjugate radii meet at M and define two angles: the sum of the two angles is 360 degrees, one angle is less than 180 degrees and the other is greater than 180 degrees. The drawing direction of the elliptical arc is the direction from the first conjugate radius to the second conjugate radius through the smaller of these two angles. Valid values of the three specifying points of the ellipse are those which yield three distinct points. The specified ellipse is non-degenerate if and only if the three points are non-colinear. Valid values of the vector components are those which produce vectors of non-zero length. If the start ray and end ray are coincident, it is ambiguous whether the defined arc is null (zero arc length) or the entire ellipse. The arc width, type and color are specified by the Line Width, Line Type and Line Color commands, respectively.

TABLE LII. Elliptical Arc input.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0 LSB	
4	18	20 = 0x4254
CenterX		
CenterY		
End1X		
End1Y		
End2X		
End2Y		
StartVectorX		
StartVectorY		
End VectorX		
EndVectorY		

5.1.1.5.6 Elliptical Arc Close element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Elliptical Arc Close element using the following format. A conjugate diameter pair of an ellipse is a pair of diameters of the ellipse such that the tangent to the ellipse at each endpoint is parallel to the other diameter. The centerpoint (CenterX, CenterY) specifies the center of an ellipse. The conjugate diameter endpoints (End1X, End1Y, and End2X, End2Y) include one endpoint from each conjugate diameter; together with the centerpoint they define the two conjugate diameters of the ellipse. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors, respectively. The define arc begins at the intersection of the ellipse and the end ray in the direction defined as follows. A "conjugate radius" is defined to be half of a conjugate diameter. Letting the centerpoint be labeled M, the first conjugate diameter endpoint P_1, and the second conjugate diameter endpoint P_2, then the line segments M-P_1 and M-P_2 define two conjugate radii, referred to in what follows as the first conjugate radius and the second conjugate radius, respectively. The conjugate radii meet at M and define two angles: the sum of the two angles is 360 degrees, one angle is less than 180 degrees and the other is greater that 180 degrees. The drawing direction of the elliptical arc is the direction from the first conjugate radius to the second conjugate radius through the smaller of these two angles. Valid values of the three specifying points of the ellipse are those which yield three distinct points. The specified ellipse is non-degenerate if and only if the three points are non-colinear. Valid values of the

vector components are those which produce vectors of non-zero length. If the start ray and end ray are coincident, it is ambiguous whether the defined arc is null (zero arc length) or the entire ellipse. The last parameter, Close Type, specifies how the arc is closed (0=pie or 1=chord). If the close type is chord, a line is drawn between the endpoints of the arc. If the close type is pie, a line is drawn from the beginning of the arc to the centerpoint of the ellipse and then to the end of the arc. The interior of the arc will be specified by the Interior Style command, and the edge width, type and color will be specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE LIII. Elliptical Arc Close input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				19							22					= 0x4276
CenterX																
CenterY																
End1X																
End1Y																
End2X																
End2Y																
StartVectorX																
StartVectorY																
End VectorX																
EndVectorY																
Close Type																

5.1.1.5.7 Rectangle element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Rectangle element using the following format. The rectangle is defined by two distinct points that are diagonal opposite corners of the rectangle where the rectangle is oriented parallel to the VDC axes. The interior of the rectangle will be as specified by the Interior Style command, and the edge width, type and color will be as specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE LIV. Rectangle input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				11							8					= 0x4168
Corner1X																
Corner1Y																
Corner2X																
Corner2Y																

5.1.1.5.8 Circle element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Circle element using the following format. The first two parameters, CenterX and CenterY, give the location of the center of the circle. The next parameter, Radius, specifies the radius of the circle. Only non-negative values are valid for the radius. The interior of the circle will be specified by the Interior Style command, and the edge width, type and color will be as specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE LV. Circle input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				12							6					= 0x4186
CenterX																
CenterY																
Radius																

5.1.1.5.9 Circular Arc Center element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Circular Arc Center element using the following format. The specified radius (Radius) and centerpoint (CenterX, CenterY) define a circle. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. The start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors respectively. The arc is drawn from the intersection of the circle and the start ray to the intersection of the circle, and the end ray in the positive angular direction, as defined by the VDC Extent. Valid values of the vector components are those which produce distinct vectors on non-zero length. The arc width, type and color are specified by the Line Width, Line Type and Line Color commands, respectively.

TABLE LVI. Circular Arc Center input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				15							14					= 0x41EE
CenterX																
CenterY																
StartX																
StartY																
EndX																
EndY																
Radius																

5.1.1.5.10 Circular Arc Center Close element input. The CGM implementation for NITFS shall provide the capability to input and interpret the Circular Arc Center Close element using the following format. The specified radius (Radius) and centerpoint (CenterX, CenterY) define a circle. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. The start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors, respectively. The arc is drawn from the intersection of the circle and the start ray to the intersection of the circle and the end ray in the positive angular direction, as defined by the VDC Extent. The last parameter, CloseType, specifies how the arc is closed (0=pie or 1=chord). If the close type is chord, a line is drawn between the endpoints of the arc. If the close type is pie, a line is drawn from the starting point through the computed arc center to the ending point. Valid values of the vector components are those which produce distinct vectors of non-zero length. The interior of this element will be as specified by the Interior Style command, and the edge width, type and color will be specified in the Edge Width, Edge Type and Edge Color commands, respectively.

TABLE LVII. Circular Arc Center Close input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				16							16					= 0x4210
CenterX																
CenterY																
StartX																
StartY																
EndX																
EndY																
Radius																
Close Type																

5.1.1.6 Control elements. The following control elements are used to describe the visual effects of auxiliary color and transparency.

5.1.1.6.1 Auxiliary Color. The CGM implementation for NITFS shall provide the capability to input and interpret the Auxiliary Color element using the following format. The Auxiliary Color element is used in

conjunction with the LINE TYPE, EDGE TYPE, and TEXT. Th Red Green, and Blue (RGB) values are specified using a single byte. The last byte of the command shall be a null byte.

TABLE LVIII. Auxiliary Color input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
3				3							3					= 0x3063
RED								GREEN								
BLUE								0								

5.1.1.6.2 Transparency. The CGM implementation for NITFS shall provide the capability to input and interpret the Transparency element using the following format. The TRANSPARENCY parameter can be off (0) or on (1).

TABLE LIX. Transparency input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
3				4							2					= 0x3082
TRANSPARENCY																

5.1.1.7 CGM binary encoding. The CGM elements are represented in the binary encoding format as described in the FIPS PUB 128 document.

5.1.1.7.1 CGM binary encoding for input. The CGM implementation for NITFS shall provide the capability to input and interpret the CGM elements in the binary encoding format.

5.1.2 CGM interface output requirements.

5.1.2.1 Delimiter elements. The Delimiter elements define boundaries for significant structures within the metafile.

5.1.2.1.1 Begin Metafile element output. The CGM implementation for NITFS shall provide the capability to generate and output the Begin Metafile element with the following formats. The Begin Metafile element name is represented using the character string C1, C2, ... Cn with length n.

TABLE LX. Begin Metafile padded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
0				1							parameter list length				
length = n (even)								C1							
C2								⋮							
Cn								0							

TABLE LXI. Begin Metafile nonpadded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
0				1							parameter list length				
length = n (odd)								C1							
C2								⋮							
C(n-1)								Cn							

TABLE LXII. Begin Metafile padded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				1							31					= 0x003F
parameter list length																
length = n (even)								C1								
C2								⋮								
Cn								0								

TABLE LXIII. Begin Metafile nonpadded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				1							31					= 0x003F
parameter list length																
length = n (odd)								C1								
C2								⋮								
C(n-1)								Cn								

5.1.2.1.2 Begin Picture element output. The CGM implementation for NITFS shall provide the capability to generate and output the Begin Picture element with the following formats. The Begin Picture element name is represented using the character string C1, C2, ... Cn with length n.

TABLE LXIV. Begin Picture padded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
0				3							parameter list length				
length = n (even)								C1							
C2								⋮							
Cn								0							

TABLE LXV. Begin Picture nonpadded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
0				3							parameter list length				
length = n (odd)								C1							
C2								⋮							
C(n-1)								Cn							

TABLE LXVI. Begin Picture padded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				3							31					= 0x007F
parameter list length																
length = n (even)								C1								
C2								⋮								
Cn								0								

TABLE LXVII. Begin Picture nonpadded, long form output

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				3							31					= 0x007F
parameter list length																
length = n (odd)								C1								
C2								⋮								
C(n-1)								Cn								

5.1.2.1.3 Begin Picture Body element output. The CGM implementation for NITFS shall provide the capability to generate and output the Begin Picture Body element using the following format.

TABLE LXVIII. Begin Picture Body output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				4							0					= 0x0080

5.1.2.1.4 End Picture element output. The CGM implementation for NITFS shall provide the capability to generate and output the End Picture element using the following format.

TABLE LXIX. End Picture output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				5							0					= 0x00A0

5.1.2.1.5 End Metafile element output. The CGM implementation for NITFS shall provide the capability to generate and output the End Metafile element using the following format.

TABLE LXX. End Metafile output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
0				2							0					= 0x0040

5.1.2.2 Metafile Descriptor elements. The Metafile Descriptor elements describe the functional content, default conditions, and characteristics of the Metafile.

5.1.2.2.1 Metafile Version element output. The CGM implementation for NITFS shall provide the capability to generate and output the Metafile Version element (version 1) using the following format.

TABLE LXXI. Metafile Version output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				1							2					= 0x1022
1																= 0x0001

5.1.2.2.2 Metafile Description element output. The CGM implementation for NITFS shall provide the capability to generate and output the Metafile Description element using the following formats. The Metafile Description element name is represented using the character string C1, C2, ... Cn with length n.

TABLE LXXII. Metafile Description padded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				2							parameter list length				
Length = n (even)								C1							
C2								⋮							
Cn								0							

TABLE LXXIII. Metafile Description nonpadded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				2							parameter list length				
Length = n (odd)											C1				
C2											⋮				
C(n-1)											Cn				

TABLE LXXIV. Metafile Description padded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				2							31					= 0x105F
parameter list length																
length = n (even)											C1					
C2											⋮					
Cn											0					

TABLE LXXV. Metafile Description nonpadded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				2							31					= 0x105F
parameter list length																
length = n (0dd)											C1					
C2											⋮					
C(n-1)											Cn					

5.1.2.2.3 Metafile Element List element output. The CGM implementation for NITFS shall provide the capability to generate and output the Metafile Element List element using the following format. The subset of commands from which metafile elements are chosen is specified by the CGM defined as "Drawing Plus Control Set" when the second parameter is one.

TABLE LXXVI. Metafile Element List output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				11							6					= 0x1166
1																= 0x0001
-1																= 0xFFFF
1																= 0x0001

5.1.2.2.4 Font List element output. The CGM implementation for NITFS shall provide the capability to generate and output the Font List element using the following formats. The first font name defined in the list is of length x with the font name given as the character string C1, C2, Cx and assigned to index 1. The last font name defined in the list is of length z with the font name given as the character string C1, C2, ... Cz and assigned to index N. Named fonts are selected by using the Text Font Index element.

TABLE LXXVII. Font List padded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				13							parameter list length				
length = x								C1							
C2								:							
C(x-1)								Cx							
length = z								C1							
C2								:							
Cz								0							

TABLE LXXVIII. Font List nonpadded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				13							parameter list length				
Length = x								C1							
C2								:							
:								C(x-1)							
:								Length = z							
C1								C2							
:								Cz							

TABLE LXXIX. Font List padded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				13							31					= 0x11BF
parameter list length																
length = x								C1								
C2								:								
C(x-1)								Cx								
length = z								C1								
C2								:								
Cz								0								

TABLE LXXX. Font List nonpadded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
1				13							31					= 0x11BF
parameter list length																
length = n								C1								
C2								:								
:								C(x-1)								
Cx								Length = z								
C1								C2								
:								Cz								

5.1.2.3 Picture Descriptor elements. The Picture Descriptor Elements set the interpretation modes of the attribute elements.

5.1.2.3.1 Color Selection Mode element output. The CGM implementation for NITFS shall provide the capability to generate and output the Color Selection Mode element using the following format.

TABLE LXXXI. Color Selection Mode output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				2							2					= 0x2042
1																=0x0001

5.1.2.3.2 Edge Width Specification Mode element output. The CGM implementation for NITFS shall provide the capability to generate and output the Edge Width Specification Mode element using the following format. Edge width is specified in "absolute mode," which indicates that the width is specified in pixels. This command is used to support filled-area primitives.

TABLE LXXXII. Edge Width Specification Mode output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				5							2					= 0x20A2
0																=0x0000

5.1.2.3.3 Line Width Specification Mode element output. The CGM implementation for NITFS shall provide the capability to generate and output the Line Width Specification Mode element using the following format. Line width is specified in "absolute mode," which indicates that the width is specified in pixels. This command is used to support the Polyline primitive.

TABLE LXXXIII. Line Width Specification Mode output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				3							2					= 0x2062
0																=0x0000

5.1.2.3.4 VDC Extent element output. The CGM implementation for NITFS shall provide the capability to generate and output the VDC Extent element using the following format.

TABLE LXXXIV. VDC Extent output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
2				6							8					= 0x20C8
X1																
Y1																
X2																
Y2																

This element (illustrated on figures 4 and 5) defines the rectangular extent of the VDC space and the orientation of VDC space (the direction of the positive X and positive Y axes). The extent and orientation of VDC space is indicated by giving the coordinates of lower left hand and upper right hand corners of the VDC extent. The VDC extent space is a one-to-one map from the VDC extent to the source coordinates. The VDC extent origin (0,0) is mapped onto the source at the row and column location given by the SLOC field in the Symbol Subheader for a given image. The Symbol Subheader and the SLOC field are defined in the MIL-STD-2500 document. Note that X1 and X2 cannot be equal, and Y1 and Y2 cannot be equal. Note that the proper Character Orientation element must be present to ensure that CGM text is orientated right side up for a given VDC Extent element.

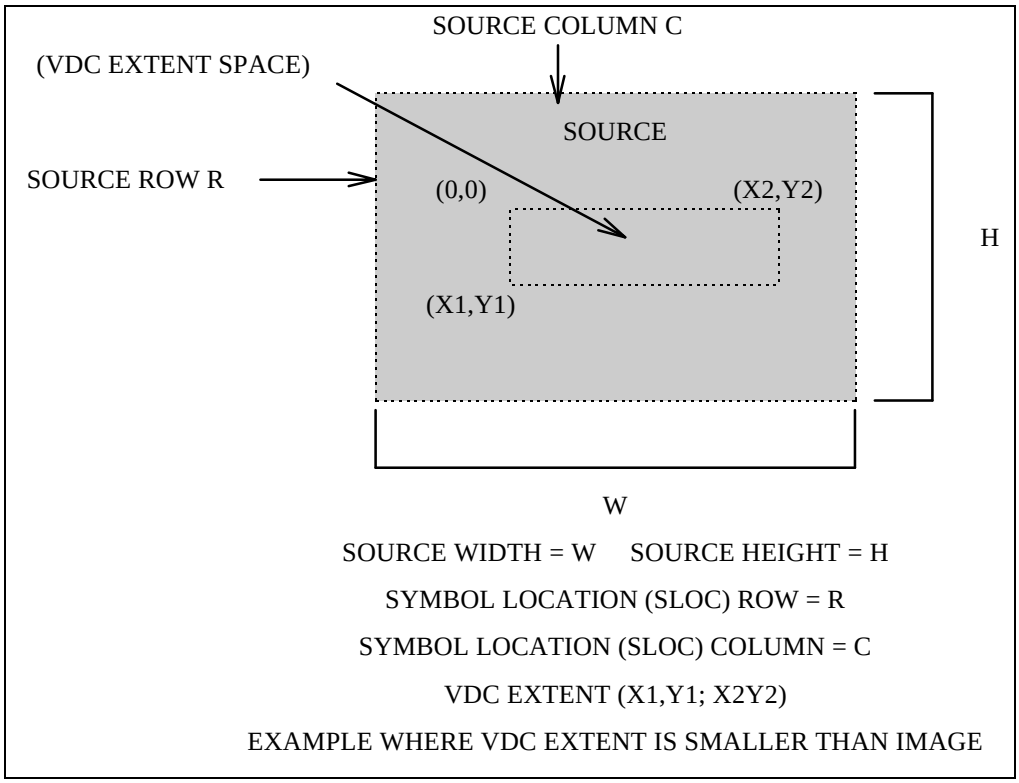

FIGURE 4. Small VDC extent output.

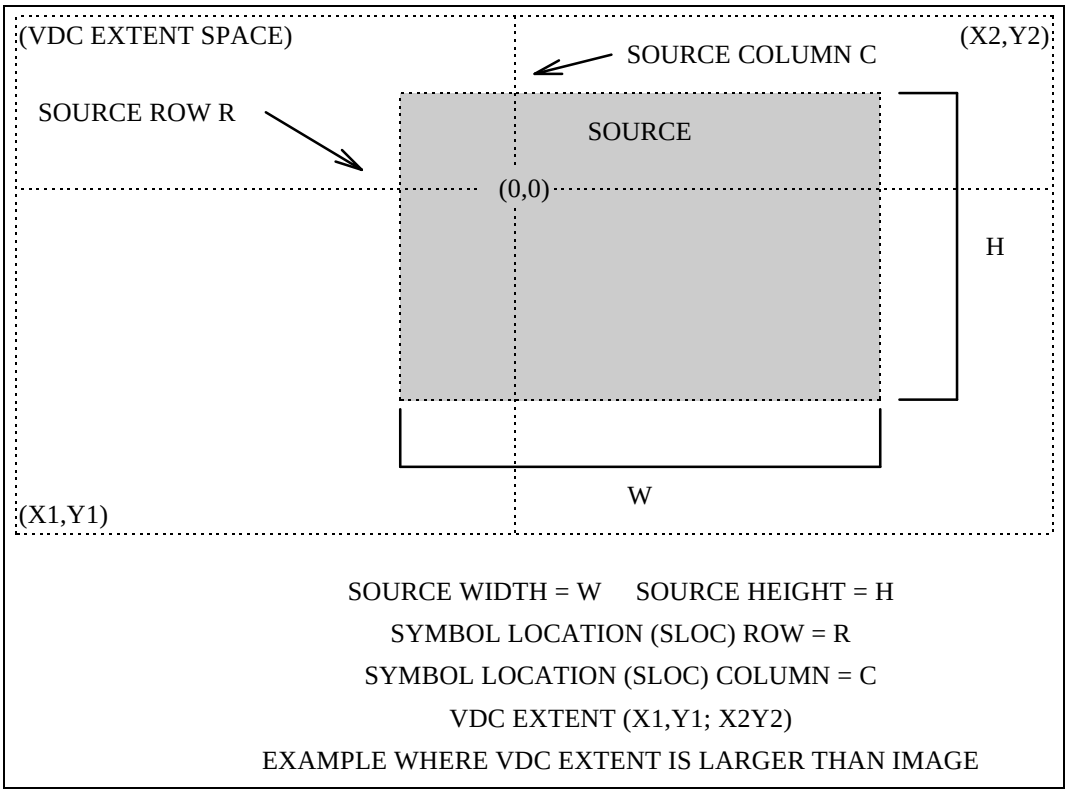

FIGURE 5. Large VDC extent output.

5.1.2.4 Attribute elements. The Attribute elements are used to describe the appearance of the Graphical Primitive elements. Full color may be specified for the Elements dealing with color. Color items for receiving systems unable to support full color must be mapped to colors they are able to support.

5.1.2.4.1 Text Color element output. The CGM implementation for NITFS shall provide the capability to generate and output the Text Color element using the following format. The Text Color element is used to support the text primitives. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE LXXXV. Text Color output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				14							3					= 0x51C3
RED								GREEN								
BLUE								0								

5.1.2.4.2 Character Height element output. The CGM implementation for NITFS shall provide the capability to generate and output the Character Height element using the following format. As measured from baseline to capline, variable character heights shall be supported. If given, the font is specified by the Font List and Text Font Index elements.

TABLE LXXXVI. Character Height output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				15							2					= 0x51E2
CHARACTER_HEIGHT																

5.1.2.4.3 Text Font Index element output. The CGM implementation for NITFS shall provide the capability to generate and output the Text Font Index element using the following format. The Text_Font_Index parameter is the index into the Font List array created from the Font List element for selection of a particular font.

TABLE LXXXVII. Text Font Index output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				10							2					= 0x5142
TEXT_FONT_INDEX																

5.1.2.4.4 Character Orientation element output. The CGM implementation for NITFS shall provide the capability to generate and output the Character Orientation element using the following format. This command defines the orientation and skew of the character body in subsequent text elements. Since the VDC Extent element is used to establish the direction of the positive X and positive Y axis, the character orientation must be present to ensure that text characters are always upright from left to right when the VDC Extent element defines Y increasing down or X increasing left. The following four cases apply to the Character Orientation element for each possible VDC extent orientation.

a. VDC Extent element with X increasing right and Y increasing up (X1<X2 and Y1<Y2).

   Character Orientation element not required or Y=1 and X=1 when Character Orientation present.

b. VDC Extent element with X increasing right and Y increasing down (X1<X2 and Y1>Y2).

   Character Orientation element required with Y=-1 and X=1.

c. VDC Extent element with X increasing left and Y increasing up (X1>X2 and Y1<Y2).

   Character Orientation element required with Y=1 and X=-1.

d. VDC Extent element with X increasing left and Y increasing down (X1>X2 and Y1>Y2).

   Character Orientation element required with Y=-1 and X=-1.

TABLE LXXXVIII. Character Orientation output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				16							8					= 0x5208
0																= 0x0000
Y																
X																
0																= 0x0000

5.1.2.4.5 Fill Color element output. Provide the capability to generate and output the Fill Color element using the following format. The Fill Color command is used to support the filled-area primitives. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE LXXXIX. Fill Color output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				23							3					= 0x52E3
RED									GREEN							
BLUE									0							

5.1.2.4.6 Interior Style element output. The CGM implementation for NITFS shall provide the capability to generate and output the Interior Style element using the following format. Filled-area primitives must support the Interior_Style parameter for solid (1), hatch (3) and empty (4).

TABLE XC. Interior Style output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				22							2					= 0x52C2
INTERIOR_STYLE																

5.1.2.4.6.1 Hatch Index output. The CGM implementation for NITFS shall provide the capability to generate and output the Hatch Index element using the following format. The Hatch Index command is used to support the primitive filled-area styles: horizontal (1), vertical (2), positive slope (3), negative slope (4), horizontal/vertical crosshatch (5) and positive/negative slope cross (6).

TABLE XC(A). Hatch Index Output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				24							2					= 0x5302
HATCH_INDEX																

5.1.2.4.7 Edge Visibility element output. The CGM implementation for NITFS shall provide the capability to generate and output the Edge Visibility element using the following format. The edge visibility command (0=off, 1=on) is supported for use in filled-area primitives.

TABLE XCI. Edge Visibility output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				30							2					= 0x52C2
EDGE_VISIBILITY																

5.1.2.4.8 Edge Width element output. The CGM implementation for NITFS shall provide the capability to generate and output the Edge Width element using the following format. Variable edge widths are supported for use in filled-area primitives.

TABLE XCII. Edge Width output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				28							2					= 0x5382
EDGE_WIDTH																

5.1.2.4.9 Edge Type element output. The CGM implementation for NITFS shall provide the capability to generate and output the Edge Type element using the following format. The Edge_Type parameter can be solid (1), dashed (2), dotted (3), dash-dot (4), and dash-dot-dot (5).

TABLE XCIII. Edge Type output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				27							2					= 0x5362
EDGE_TYPE																

5.1.2.4.10 Edge Color element output. The CGM implementation for NITFS shall provide the capability to generate and output the Edge Color element using the following format. Edge color is supported for use in filled-area primitives. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE XCIV. Edge Color output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				29							3					= 0x53A3
RED								GREEN								
BLUE								0								

5.1.2.4.11 Line Width element output. The CGM implementation for NITFS shall provide the capability to generate and output the Line Width element using the following format. Variable line widths are supported for use by line primitives.

TABLE XCV. Line Width output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				3							2					= 0x5062
LINE_WIDTH																

5.1.2.4.12 Line Type element output. The CGM implementation for NITFS shall provide the capability to generate and output the Line Type element using the following format. The Line_Type parameter can be solid (1), dashed (2), dotted (3), dash-dot (4), and dash-dot-dot (5).

TABLE XCVI. Line Type output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				2							2					= 0x5042
LINE_TYPE																

5.1.2.4.13 Line Color element output. The CGM implementation for NITFS shall provide the capability to generate and output the Line Color element using the following format. Line colors are supported for use by the line primitive. RGB values are each specified using a single byte, and the last byte of the command shall be a null byte.

TABLE XCVII. Line Color output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
5				48							3					= 0x5083
RED								GREEN								
BLUE								0								

5.1.2.5 Graphical Primitive elements. The Graphical Primitive elements describe the visual components of a picture in the CGM.

5.1.2.5.1 Text element output. The CGM implementation for NITFS shall provide the capability to generate and output the Text element using the following formats. The X and Y parameters give the coordinates of the lower left hand corner of the text string. The flag parameter is an integer (1) indicating that this is the final text in the string. The length parameter is an unsigned byte containing the number of characters in the string. Finally, the text is given as a character string C1, C2, ... Cn with length n. If necessary, the last byte is padded with a zero so that the next command begins on a word boundary.

TABLE XCVIII. Text padded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							parameter list length					
X																
Y																
1																0x0001
length = n(even)								C1								
C2								.								
Cn								0								

TABLE XCIX. Text nonpadded, short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							parameter list length					
X																
Y																
1																0x0001
length = n (odd)								C1								
C2								.								
C(n-1)								Cn								

TABLE C. Text padded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							31					0x409F
parameter list length																
X																
Y																
1																0x0001
length = n (even)								C1								
C2								.								
Cn								0								

TABLE CI. Text nonpadded, long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				4							31					0x409F
parameter list length																
X																
Y																
1																0x0001
length = n (odd)								C1								
C2								.								
C(n-1)								Cn								

5.1.2.5.2 Polygon element output. The CGM implementation for NITFS shall provide the capability to generate and output the Polygon element using the following format. The polygon parameters consist of a list of pairs of coordinates indicating the vertices of a polygon. The first vertex (Vertex1X, Vertex1Y) is connected to the last (Vertex(N)X, Vertex(N)Y) to close the polygon. Polygons are not "clipped" to the image boundary. Therefore, some coordinates may specify off-image or off-screen locations, including negative locations. Note, the parameter list length is given as the total number of bytes for all vertex parameters (4N).

TABLE CII. Polygon short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
1				7							parameter list length (4N)				
Vertex1X															
Vertex1Y															
Vertex2X															
Vertex2Y															
.															
Vertex(N)X															
Vertex(N)Y															

TABLE CIII. Polygon long form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				7							31					0x40FF
Parameter list length (4N)																
Vertex1X																
Vertex1Y																
Vertex2X																
Vertex2Y																
.																
Vertex(N)X																
Vertex(N)Y																

5.1.2.5.2.1 Polygon Set element output. The CGM implementation for NITFS shall provide the capability to generate and output the Polygon Set element using the following format. The polygon set parameters consist of a list of pairs of coordinates indicating the vertices of each polygon in the polygon set along with the Edge_Out_Flag indicating the edge visibility and whether the vertex is the last (closure) vertex of the specific polygon in the set. The first vertex of the first polygon (Vertex_11X, Vertex_11Y, Edge_Out_Flag_11) is connected to the last (Vertex_1(N)X, Vertex_1(N)Y, Edge_Out_Flag_1(N)) vertex in the first polygon followed by the vertices of each successive polygon in like manner. Polygon Sets are not “clipped” to the image boundary; therefore, some coordinates may specify off-image or off-screen locations, including negative locations. The Edge_Out_Flag parameter can be Invisible (0), Visible (1), Close Invisible (2), or Close Visible (3). Note, the

parameter list length is given as the total number of bytes for all vertex parameters (6N).

TABLE CIV. Polygon Set short form output.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0 LSB
1	8 | parameter list length (6N)
Vertex_11X	
Vertex_11Y	
Edge_Out_Flag_11	
Vertex_12X	
Vertex_12Y	
Edge_Out_Flag_12	
⋮	
Vertex_(n)(N)X	
Vertex_(n)(N)Y	
Edge_Out_Flag_(n)(N)	

TABLE CV. Polygon Set long form output.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0 LSB	
4	7 | 31	= 411F
parameter list length (6N)		
Vertex_11X		
Vertex_11Y		
Edge_Out_Flag_11		
Vertex_12X		
Vertex_12Y		
Edge_Out_Flag_12		
⋮		
Vertex_1(N)X		
Vertex_1(N)Y		
Edge_Out_Flag_1(N)		
Vertex_21X		
Vertex_21Y		
Edge_Out_Flag_21		
Vertex_22X		
Vertex_22Y		
Edge_Out_Flag_22		
⋮		
Vertex_2(N)X		
Vertex_2(N)Y		
Edge_Out_Flag_2(N)		
⋮		
Vertex_(n)1X		
Vertex_(n)1Y		
Edge_Out_Flag_(n)1		
Vertex_(n)2X		
Vertex_(n)2Y		
Edge_Out_Flag_(n)2		

TABLE CV. Polygon Set long form output - Continued.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				7							31					= 411F
Vertex_(n)(N)X																
Vertex_(n)(N)Y																
Edge_Out_Flag_(n)(N)																

5.1.2.5.3 Ellipse element output. The CGM implementation for NITFS shall provide the capability to generate and output the Ellipse element using the following format. The ellipse parameters consist of a list of three pairs of coordinates. The first pair, CenterX and CenterY, give the location of the center of the ellipse. The other two pairs, End1X, End1Y, and End2X, End2Y, specify the end points of the first and second conjugate diameters. The ellipse interior will be as specified by the Interior Style command, and the edge width, type, and color will be as specified in the Edge Width, Edge Type, and Edge Color commands respectively.

TABLE CVI. Ellipse output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				17							12					0x422C
CenterX																
Center Y																
End1X																
End1Y																
End2X																
End2Y																

5.1.2.5.4 Polyline element output. The CGM implementation for NITFS shall provide the capability to generate and output the Polyline element using the following format. The polyline parameters consist of a list of pairs of coordinates indicating the vertices of a polyline (Vertex1X, Vertex1Y) to (Vertex(N)X, Vertex(N)Y). The line width, type, and color are specified by the Line Width, Line Type, and Line Color commands, respectively. Polylines are not "clipped" to the image boundary. Therefore, some coordinates may specify off-image or off-screen locations, including negative locations. Note, the parameter list length is given as the total number of bytes for all vertex parameters (4N).

TABLE CVII. Polyline short form output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
4				1							parameter list length (4N)				
Vertex1X															
Vertex1Y															
Vertex2X															
Vertex2Y															
.															
Vertex(N)X															
Vertex(N)Y															

TABLE CVIII. Polyline long form output.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0 LSB	
4	1	31	0x403F
parameter list length (4N)		
Vertex1X		
Vertex1Y		
Vertex2X		
Vertex2Y		
.		
Vertex(N)X		
Vertex(N)Y		

5.1.2.5.5 Elliptical Arc element output. The CGM implementation for NITFS shall provide the capability to generate and output the Elliptical Arc element using the following format. A conjugate diameter pair of an ellipse is a pair of diameters of the ellipse such that a tangent to the ellipse at each endpoint is parallel to the other diameter. The centerpoint (CenterX, CenterY) specifies the center of an ellipse. The conjugate diameter endpoints (End1X, End1Y, and End2X, End2Y) include one endpoint from each conjugate diameter. Together with the centerpoint, they define the two conjugate diameters of the ellipse. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors, respectively. The define arc begins at the intersection of the ellipse and the end ray in the direction defined as follows. A "conjugate radius" is defined to be half of a conjugate diameter. Letting the centerpoint be labeled M, the first conjugate diameter endpoint P_1, and the second conjugate diameter endpoint P_2, then the line segments M-P_1 and M-P_2 define two conjugate radii, referred to in what follows as the first conjugate radius and the second conjugate radius, respectively. The conjugate radii meet at M and define two angles: the sum of the two angles is 360 degrees, one angle is less than 180 degrees and the other is greater than 180 degrees. The drawing direction of the elliptical arc is less than 180 degrees and the other is greater than 180 degrees. The drawing direction of the elliptical arc is the direction from the first conjugate radius to the second conjugate radius through the smaller of these two angles. Valid values of the three specifying points of the ellipse are those which yield three distinct points. The specified ellipse is non-degenerate if and only if the three points are non-colinear. Valid values of the vector components are those which produce vectors of non-zero length. If the start ray and end ray are coincident, it is ambiguous whether the defined arc is null (zero arc length) or the entire ellipse. The arc width, type and color are specified by the Line Width, Line Type, and Line Color commands, respectively.

TABLE CIX. Elliptical Arc output.

MSB 15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0 LSB	
4	18	20	0x4254
CenterX		
CenterY		
End1X		
End1Y		
End2X		
End2Y		
StartVectorX		
StartVentorY		
EndVectorX		
EndVectorY		

5.1.2.5.6 Elliptical Arc Close element output. The CGM implementation for NITFS shall provide the capability to generate and output the Elliptical Arc Close element using the following format. A conjugate diameter pair of an ellipse is a pair of diameters of the ellipse such that a tangent to the ellipse at each endpoint is

parallel to the other diameter. The centerpoint (CenterX, CenterY) specifies the center of an ellipse. The conjugate diameter endpoints (End1X, End1Y, and End2X, End2Y) include one endpoint from each conjugate diameter. Together with the centerpoint, they define the two conjugate diameters of the ellipse. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors, respectively. The define arc begins at the intersection of the ellipse and the end ray in the direction defined as follows. A "conjugate radius" is defined to be half of a conjugate diameter. Letting the centerpoint be labeled M, the first conjugate diameter endpoint P_1, and the second conjugate diameter endpoint P_2, then the line segments M-P_1 and M-P_2 define two conjugate radii, referred to in what follows as the first conjugate radius and the second conjugate radius, respectively. The conjugate radii meet at M and define two angles: the sum of the two angles is 360 degrees, one angle is less than 180 degrees and the other is greater than 180 degrees. The drawing direction of the elliptical arc is the direction from the first conjugate radius to the second conjugate radius through the smaller of these two angles. Valid values of the three specifying points of the ellipse are those which yield three distinct points. The specified ellipse is non-degenerate if and only if the three points are non-colinear. Valid values of the vector components are those which produce vectors of non-zero length. If the start ray and end ray are coincident, it is ambiguous whether the defined arc is null (zero arc length) or the entire ellipse. The last parameter, Close Type, specifies how the arc is closed (0=pie or 1=chord). If the close type is chord, a line is drawn between the endpoints of the arc. If the close type is pie, a line is drawn from the beginning of the arc to the centerpoint of the ellipse and then to the end of the arc. The interior of the arc will be as specified by the Interior Style command, and the edge width, type and color will be specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE CX. Elliptical Arc output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				19							22					0x4276
CenterX																
CenterY																
End1X																
End1Y																
End2X																
End2Y																
StartVectorX																
StartVentorY																
EndVectorX																
EndVectorY																
Close Type																

5.1.2.5.7 Rectangle element output. The CGM implementation for NITFS shall provide the capability to generate and output the Rectangle element using the following format. The rectangle is defined by two distinct points that are diagonally opposite corners of the rectangle where the rectangle is oriented parallel to the VDC axes. The interior of the rectangle will be as specified in the Interior Style command, and the edge width, type and color will be specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE CXI. Rectangle output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				11							8					0x4168
Corner1X																
Corner1Y																
Corner2X																
Corner2Y																

5.1.2.5.8 Circle element output. The CGM implementation for NITFS shall provide the capability to generate and output the Circle element using the following format. The first two parameters, CenterX and

CenterY, give the location of the center of the circle. Only non-negative values are valid for the radius. The interior of the circle will be as specified in the Interior Style command, and the edge width, type and color will be specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE CXII. Circle output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				12							6					0x4186
CenterX																
CenterY																
Radius																

5.1.2.5.9 Circular Arc Center element output. The CGM implementation for NITFS shall provide the capability to generate and output the Circular Arc Center element using the following format. The specified radius (Radius) and centerpoint (CenterX, CenterY) define a circle. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. The start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors, respectively. The arc is drawn from the intersection of the circle and the start ray to the intersection of the circle and the end ray in the positive angular direction, as defined by the VDC Extent. Valid values of the vector components are those which produce distinct vectors of non-zero length. The arc width, type and color are specified by the Line Width, Line Type, and Line Color commands, respectively.

TABLE CXIII. Circular Arc Center output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				15							14					0x41EE
CenterX																
CenterY																
StartX																
StartY																
EndX																
EndY																
Radius																

5.1.2.5.10 Circular Arc Center Close element output. The CGM implementation for NITFS shall provide the capability to generate and output the Circular Arc Center Close element using the following format. The specified radius (Radius) and centerpoint (CenterX, CenterY) define a circle. StartVectorX and StartVectorY define a start vector, and EndVectorX and EndVectorY define an end vector. The tails of these vectors are placed on the centerpoint. The start ray and end ray are derived from the start and end vectors. The start and end rays are the semi-infinite lines from the centerpoint in the direction of the start and end vectors, respectively. The arc is drawn from the intersection of the circle and the start ray to the intersection of the circle and the end ray in the positive angular direction, as defined by the VDC Extent. The last parameter, CloseType, specifies how the arc is closed (0=pie or 1=chord). If the close type is chord, a line is drawn between the endpoints of the arc. If the close type is pie, a line is drawn from the starting point through the computed arc center to the ending point. Valid values of the vector components are those which produce distinct vectors of non-zero length. The interior of this element will be as specified by the Interior Style command, and the edge width, type and color will be specified in the Edge Width, Edge Type, and Edge Color commands, respectively.

TABLE CXIV. Circular Arc Center Close output.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
4				16							16					0x4210
CenterX																
CenterY																
StartX																
StartY																
EndX																
EndY																
Radius																
Close Type																

5.1.2.6 Control elements. The following control elements are used to describe the visual effects of auxiliary color and transparency.

5.1.2.6.1 Auxiliary Color. The CGM implementation for NITFS shall provide the capability to generate and output the Auxiliary Color element using the following format. The Auxiliary Color element is used in conjunction with the LINE TYPE, EDGE TYPE, and TEXT. The Red, Green, and Blue (RGB) values are specified using a single byte. The last byte of the command shall be a null byte.

TABLE CXV. Auxiliary Color input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
3				3							3					0x3063
RED								GREEN								
BLUE								0								

5.1.2.6.2 Transparency. The CGM implementation for NITFS shall provide the capability to generate and output the Transparency element using the following format. The TRANSPARENCY parameter can be off (0) or on (1).

TABLE CXVI. Transparency input.

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0	
3				4							2					0x3082
TRANSPARENCY																

5.1.2.7 CGM binary encoding. The CGM elements are represented in the binary encoding format as described in the FIPS PUB 128 document.

5.1.2.7.1 CGM binary encoding for output. The CGM implementation for NITFS shall provide the capability to generate and output the CGM elements in the binary encoding format.

5.2 Functional requirements. The CGM implementation for NITFS complies with the specifications established in FIPS PUB 128. The following requirements establish the element position and functional requirements for the CGM implementation for NITFS. The words "command" and "element" are used synonymously throughout this document.

5.2.1 CGM element position requirements.

5.2.1.1 CGM element position input requirements.

5.2.1.1.1 Begin Metafile element position for input. The CGM implementation for NITFS shall interpret the Begin Metafile element as the first command in the metafile.

5.2.1.1.2 Begin Picture element position for input. The CGM implementation for NITFS shall interpret the Begin Picture element after all Metafile Descriptor elements in the metafile.

5.2.1.1.3 Begin Picture Body element position for input. The CGM implementation for NITFS shall interpret the Begin Picture Body element after each Begin Picture element and its associated Picture Descriptor elements.

5.2.1.1.4 End Picture element position for input. The CGM implementation for NITFS shall interpret the End Picture element after any Begin Picture Body element and its associated Attribute and Graphical Primitive elements.

5.2.1.1.5 End Metafile element position for input. The CGM implementation for NITFS shall interpret the End Metafile element as the last element in the metafile.

5.2.1.1.6 Metafile Descriptor elements position for input. The CGM implementation for NITFS shall interpret all Metafile Descriptor elements after the Begin Metafile element and before any Begin Picture elements. The Metafile Version element, the Metafile Element List element and the Metafile Description element shall be the first three metafile descriptor elements in that order.

5.2.1.1.7 Picture Descriptor elements position for input. The CGM implementation for NITFS shall interpret all Picture Descriptor elements after the Begin Picture element and before the Begin Picture Body element.

5.2.1.1.8 Attribute elements position for input. The CGM implementation for NITFS shall interpret any Attribute Element after the Begin Picture Body element and before the Graphical Primitive element to which it applies.

5.2.1.1.9 Graphical Primitive elements position for input. The CGM implementation for NITFS shall interpret any Graphical Primitive elements after the Begin Picture Body element and any associated Attribute elements for that Graphical Primitive element and before the End Picture element.

5.2.1.2 CGM element position output requirements.

5.2.1.2.1 Begin Metafile element position for output. The CGM implementation for NITFS shall generate the Begin Metafile element as the first command in the metafile.

5.2.1.2.2 Begin Picture element position for output. The CGM implementation for NITFS shall generate the Begin Picture element after all Metafile Descriptor elements in the metafile.

5.2.1.2.3 Begin Picture Body element position for output. The CGM implementation for NITFS shall generate the Begin Picture Body element after each Begin Picture element and its associated Picture Descriptor elements.

5.2.1.2.4 End Picture element position for output. The CGM implementation for NITFS shall generate the End Picture element after any Begin Picture Body element and its associated Attribute and Graphical Primitive elements.

5.2.1.2.5 End Metafile element position for output. The CGM implementation for NITFS shall generate the End Metafile element as the last element in the metafile.

5.2.1.2.6 Metafile Descriptor elements position for output. The CGM implementation for NITFS shall generate all Metafile Descriptor elements after the Begin Metafile element and before any Begin Picture elements. The Metafile Version element, the Metafile Element List element, and the Metafile Description element shall be the first three metafile descriptor elements in that order.

5.2.1.2.7 Picture Descriptor elements position for output. The CGM implementation for NITFS shall generate all Picture Descriptor elements after the Begin Picture element and before the Begin Picture Body element.

5.2.1.2.8 Attribute elements position for output. The CGM implementation for NITFS shall generate any Attribute Element after the Begin Picture Body element and before the Graphical Primitive element to which it applies.

5.2.1.2.9 Graphical Primitive elements position for output. The CGM implementation for NITFS shall generate any Graphical Primitive elements after the Begin Picture Body element and any associated Attribute elements for that Graphical Primitive element and before the End Picture element.

5.2.2 CGM element functional requirements.

5.2.2.1 CGM element functional input requirements.

5.2.2.1.1 CGM input required elements. The following CGM elements are required for each CGM that the CGM implementation for NITFS inputs and interprets:

a. Begin Metafile

b. Metafile Version

c. Metafile Element List

d. Metafile Description

e. Begin Picture

f. Color Selection Mode

g. VDC Extent

h. Begin Picture Body

i. End Picture

j. End Metafile.

5.2.2.1.2 Metafile Description element contents required for input. The CGM implementation for NITFS shall be able to input and interpret the Metafile Description element that contains the following substring:

NITF/CGM-APP-2.0 for CGM placed in NITF2.0 and/or NITF2.1 files.

And additionally, for NITF2.1 files only:

NITF/CGM;ProfileEd:2301-2/Source:*producer*;Date:YYYYMMDD”

Where ‘*producer*’ is the application name and release/version of the application producing or modifying the CGM. The date shall reflect the creation date or more recent date of modification.

5.2.2.1.3 Length of parameter strings required for input for the Begin Metafile, Begin Picture, and Metafile Description elements. The CGM implementation for NITFS shall be able to input and interpret the Begin Metafile, Begin Picture, and Metafile Description elements with parameter strings of at least 1024 characters.

5.2.2.1.4 Length of parameter strings required for input for the Font List element. The CGM implementation for NITFS shall be able to input and interpret the Font List element with parameter strings of at least 1024 characters.

5.2.2.1.5 Number of Begin Picture elements and Begin Picture Body elements required for input. The CGM implementation for NITFS shall be able to input and interpret one Begin Picture element with only one corresponding Begin Picture Body element.

5.2.2.1.6 End Picture element required for input. The CGM implementation for NITFS shall be able to input and interpret a CGM where an End Picture element occurs for each Begin Picture element.

5.2.2.1.7 VDC Extent element required for input. The CGM implementation for NITFS shall provide the capability to input and interpret a CGM where the VDC extent origin maps to the row and column given in the SLOC field in the National Imagery Transmission Format (NITF) symbol subheader and the VDC extent space maps one-to-one to the source coordinates.

5.2.2.1.8 Edge Width Specification Mode element for input. The CGM implementation for NITFS shall be able to input and interpret a CGM where the Edge Width Specification Mode element occurs before any filled-area primitive element. The representation of edge widths shall be centered (within plus or minus one pixel) on the ideal mathematically-defined edge of the area**.**

5.2.2.1.9 Line Width Specification Mode element for input. The CGM implementation for NITFS shall be able to input and interpret a CGM where the Line Width Specification Mode element occurs before any line primitive element. The representation of line widths shall be centered (within plus or minus one pixel) on the ideal mathematically-defined center of the line**.**

5.2.2.1.10 Color Selection Mode element required for input. The CGM implementation for NITFS shall be able to input and interpret a CGM where the Color Selection Mode element occurs before any graphics primitive.

5.2.2.1.11 Character Orientation element required for input. The CGM implementation for NITFS shall be able to input and interpret a CGM where the character orientation element occurs before the Text Primitive element to ensure that the text is displayed upright and left to right.

5.2.2.1.12 Font List number required for input. The CGM implementation for NITFS shall be able to input and interpret at least 32 font name entries in the Font List element.

5.2.2.1.13 Font names for input. The CGM implementation for NITFS shall use a system font for any unsupported font name specified in the Font List element. The substituted system fonts shall have similar visual and metric characteristics to the font specified in the metafile.

5.2.2.1.14 Text Font Index required for input. The CGM implementation for NITFS shall be able to input and interpret a Text Font Index element with index n when the Font List element has been interpreted with the number of fonts at least n.

5.2.2.1.15 Edge widths for input. The CGM implementation for NITFS shall be able to substitute default system edge widths for any unsupported edge widths from the Edge Width element.

5.2.2.1.16 Line widths for input. The CGM implementation for NITFS shall be able to substitute default system line widths for any unsupported line widths from the Line Width element.

5.2.2.1.17 Edge types required for input. The CGM implementation for NITFS shall be able to substitute solid edge type for any unsupported edge type from the Edge Type element.

5.2.2.1.18 Line types required for input. The CGM implementation for NITFS shall be able to substitute solid line type for any unsupported line type from the Line Type element.

5.2.2.1.19 Interior styles required for input. The CGM implementation for NITFS shall be able to substitute empty interior style for any unsupported interior style from the Interior Style element.

5.2.2.1.20 Text element requirements for input. The CGM implementation for NITFS shall be able to input and interpret the Text element with text string parameter length at least 254 characters.

5.2.2.1.21 Character Height requirements for input. The CGM implementation for NITFS shall be able to substitute default system text heights for any unsupported character height in the Character Height element. The implementation shall at least support character heights within the range of 6 through 72, although not all values within the range need to be supported for each font. When receiving an unsupported character height, the substituted height shall be the next lowest supported value for the font.

5.2.2.1.22 Polyline element requirements for input. The CGM implementation for NITFS shall be able to input and interpret the Polyline element with at least 4096 vertices.

5.2.2.1.23 Polygon and Polygon Set element requirements for input. The CGM implementation for NITFS shall be able to input and interpret the Polygon and Polygon Set element with at least 4096 vertices.

5.2.2.1.24 Input and interpret in sequential order. The CGM implementation for NITFS shall be able to interpret the CGM elements in the sequential order from which they are input from the metafile.

5.2.2.1.25 Input Text element. The CGM implementation for NITFS shall be able to input and interpret the Text element using the following attributes.

a. Text Color

b. Character Height

c. Text Font Index

d. Character Orientation

5.2.2.1.26 Input Polygon and Polygon Set element. The CGM implementation for NITFS shall be able to input and interpret the Polygon element using the following attributes.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.1.27 Input Ellipse element. The CGM implementation for NITFS shall be able to input and interpret the Ellipse element using the following attributes.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.1.28 Input Polyline element. The CGM implementation for NITFS shall be able to input and interpret the Polyline element using the following attributes.

a. Line Width

b. Line Type

c. Line Color

5.2.2.1.29 Input Elliptical Arc element. The CGM implementation for NITFS shall be able to input and interpret the Elliptical Arc element using the following attributes.

a. Line Width

b. Line Type

c. Line Color

5.2.2.1.30 Input Elliptical Closed Arc element. The CGM implementation for NITFS shall be able to input and interpret the Elliptical Arc Close element using the following attributes.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.1.31 Input Rectangle element. The CGM implementation for NITFS shall be able to input and interpret the Rectangle element using the following attributes.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.1.32 Input Circle element. The CGM implementation for NITFS shall be able to input and interpret the Circle element using the following attributes.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.1.33 Input Circular Arc Center element. The CGM implementation for NITFS shall be able to input and interpret the Circular Arc Center element using the following attributes.

a. Line Width

b. Line Type

c. Line Color

5.2.2.1.34 Input Circular Arc Center Close element. The CGM implementation for NITFS shall be able to input and interpret the Circular Arc Center Close element using the following attributes.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.1.35 Auxiliary Color for input. The CGM implementation for NITFS shall be able to input and interpret the Auxiliary Color element when used in conjunction with EDGE TYPE, LINE TYPE, and TEXT.

5.2.2.1.36 Transparency for input. The CGM implementation for NITFS shall be able to input and interpret the Transparency element when used in conjunction with EDGE TYPE, LINE TYPE, and TEXT.

5.2.2.1.37 CGM element defaults for input. The CGM implementation for NITFS shall assume all CGM default values as stated in the Part 3 - Binary Encoding in the FIPS PUB 128 for each CGM that the CGM implementation inputs and interprets when these elements are not expressly included in the CGM unless otherwise specified in this document. The CGM element defaults for input include, but are not limited to those listed in table CXVII.

TABLE CXVII. CGM element defaults for input.

VDC TYPE:	16 BIT INTEGER
INTEGER PRECISION:	16 BIT INTEGER
INDEX PRECISION:	16 BIT INTEGER
COLOR PRECISION:	8 BIT INTEGER
TRANSPARENCY:	ON
LINE TYPE:	1 (SOLID)
TEXT PRECISION:	STRING
CHARACTER EXPANSION FACTOR:	1.0
CHARACTER SPACING:	0.0
CHARACTER ORIENTATION:	0, 1, 1, 0
TEXT PATH:	RIGHT
TEXT ALIGNMENT:	NORMAL HORIZONTAL, NORMAL VERTICAL
INTERIOR STYLE:	HOLLOW (EMPTY)
EDGE TYPE:	SOLID
EDGE VISIBILITY:	OFF
LINE COLOR:	DEVICE-DEPENDENT FOREGROUND COLOR
EDGE COLOR:	DEVICE-DEPENDENT FOREGROUND COLOR
FILL COLOR:	DEVICE-DEPENDENT FOREGROUND COLOR
TEXT COLOR:	DEVICE-DEPENDENT FOREGROUND COLOR
BACKGROUND COLOR:	NONE (THIS IS NITF SPECIFIC)
COLOR VALUE EXTENT:	0,0,0 - 255,255,255
VDC INTEGER PRECISION:	16 BIT INTEGER
TEXT FONT INDEX:	1
HATCH INDEX	Horizontal (1)

5.2.2.1.38 Default colors for unsupported text. The CGM implementation for NITFS shall substitute available system colors for unsupported colors specified in the Text Color element, Fill Color element, Edge Color element, and the Line Color element.

5.2.2.1.39 CGM element substitution. The CGM implementation for NITFS shall report or substitute for any CGM element and associated parameters not supported in the CGM implementation for NITFS and continue to interpret the next element supported in the CGM implementation for the NITFS.

5.2.2.1.40 CGM error messages. The CGM implementation for NITFS should report errors encountered during the input and interpretation of the CGM.

5.2.2.2 CGM element functional output requirements.

5.2.2.2.1 CGM output required elements. The following CGM elements are required for each CGM that the CGM implementation for NITFS generates and outputs:

a. Begin Metafile

b. Metafile Version

c. Metafile Element List

d. Metafile Description

e. Begin Picture

f. Color Selection Mode

g. VDC Extent

h. Begin Picture Body

i. End Picture

j. End Metafile

5.2.2.2.2 Metafile Description element contents required for output. The CGM implementation for NITFS shall generate and output the Metafile Description element that contains the following substring:

NITF/CGM-APP-2.0 for CGM placed in NITF2.0 and/or NITF 2.1 files.

And additionally, for NITF 2.1 files only:

NITF/CGM;ProfileEd:2301-2/Source:*producer*;Date:YYYYMMDD

Where '*producer*' is the application name and release/version of the application producing or modifying the CGM. The date shall reflect the creation date or more recent date of modification.

5.2.2.2.3 Length of parameter strings required for output for the Begin Metafile, Begin Picture, and Metafile Description elements. The CGM implementation for NITFS shall generate and output the Begin Metafile, Begin Picture, and Metafile Description elements with parameter strings not to exceed 1024 characters.

5.2.2.2.4 Length of parameter strings required for output for the Font List element. The CGM implementation for NITFS shall generate and output the Font List element with parameter strings not to exceed 1024 characters.

5.2.2.2.5 Number of Begin Picture elements and Begin Picture Body elements required for output. The CGM implementation for NITFS shall generate and output only one Begin Picture element with only on Begin Picture Body element.

5.2.2.2.6 End Picture element required for output. Exactly one End Picture element is required for each Begin Picture element for each CGM that the CGM implementation for NITFS generates and outputs.

5.2.2.2.7 Edge Width Specification Mode element for output. The Edge Width Specification Mode element is required when any filled-area primitive element is present for each CGM that the CGM implementation for NITFS generates and outputs. The representation of edge widths shall be centered (within plus or minus one pixel) on the ideal mathematically-defined edge of the area**.**

5.2.2.2.8 Line Width Specification Mode element for output. The Line Width Specification Mode element is required when any line primitive element is present for each CGM that the CGM implementation for NITFS generates and outputs. The representation of line widths shall be centered (within plus or minus one pixel) on the ideal mathematically-defined center of the line**.**

5.2.2.2.9 Color Selection Mode element required for output. The Color Selection Mode element is required for each CGM that the CGM implementation for NITFS generates and outputs.

5.2.2.2.10 Character Orientation element required for output. The character orientation element is required for the Text Primitive element when the VDC Extent element parameters are given as X1 > X2 or Y1 > Y2. This ensures that the text is displayed upright and left to right for each CGM that the CGM implementation for NITFS generates and outputs.

5.2.2.2.11 Font List number required for output. The CGM implementation for NITFS shall provide the capability to generate and output, at most, 32 font name entries in the Font List element.

5.2.2.2.12 Font names for output. The CGM implementation for NITFS shall limit the font name in the Font List element to the following list. The HERSHEY fonts are specified in NBS SP 424 while TIMES, COURIER and HELVETICA fonts are registered trademarks of Allied Corporation which owns the copyright.

a. HERSHEY/CARTOGRAPHIC_ROMAN

b. HERSHEY/CARTOGRAPHIC_GREEK

c. HERSHEY/SIMPLEX_ROMAN

d. HERSHEY/SIMPLEX_GREEK

e. HERSHEY/SIMPLEX_SCRIPT

f. HERSHEY/COMPLEX_ROMAN

g. HERSHEY/COMPLEX_GREEK

h. HERSHEY/COMPLEX_SCRIPT

i. HERSHEY/COMPLEX_ITALIC

j. HERSHEY/COMPLEX_CYRILLIC

k. HERSHEY/DUPLEX_ROMAN

l. HERSHEY/TRIPLEX_ROMAN

m. HERSHEY/TRIPLEX_ITALIC

n. HERSHEY/GOTHIC_GERMAN

o. HERSHEY/GOTHIC_ENGLISH

p. HERSHEY/GOTHIC_ITALIAN

q. TIMES_ROMAN

r. TIMES_ITALIC

s. TIMES_BOLD

t. TIMES_BOLD_ITALIC

u. HELVETICA

v. HELVETICA_OBLIQUE

w. HELVETICA_BOLD

x. HELVETICA_BOLD_OBLIQUE

y. COURIER

z. COURIER_BOLD

aa. COURIER_ITALIC

ab. COURIER_BOLD_ITALIC

5.2.2.2.13 Text Font Index required for output. The index of every Text Font Index element shall be less than or equal to (≤) the number of fonts specified in the Font List element for any CGM implementation generated for NITFS. The Font List element is required for each CGM that contains a Text Font Index element.

5.2.2.2.14 Edge widths for output. The CGM implementation for NITFS shall provide the capability to generate and output edge widths of two, four, or six**,** and optionally, of 1 through 100 for the Edge Width element.

5.2.2.2.15 Line widths for output. The CGM implementation for NITFS shall provide the capability to generate and output line widths of two, four, or six**,** and optionally, of 1 through 100 for the Line Width element.

5.2.2.2.16 Edge types required for output. The CGM implementation for NITFS shall generate and output edge types of solid (1)**,** dashed (2), dotted (3), dash-dot (4), and dash-dot-dot (5), to the Edge Type element.

5.2.2.2.17 Line types required for output. The CGM implementation for NITFS shall generate and output line types of solid (1)**,** dashed (2), dotted (3), dash-dot (4), and dash-dot-dot (5), to the Line Type element.

5.2.2.2.18 Interior styles required for output. The CGM implementation for NITFS shall generate and output interior styles of solid (1), hatch (3), empty (4), to the Interior Style element.

5.2.2.2.19 Text element requirements for output. The CGM implementation for NITFS shall generate and output the Text element with text string parameter not to exceed 254 characters.

5.2.2.2.20 Character Height requirements for output. The CGM implementation for NITFS shall generate and output the Character Height element with a value no less than 6. There is no constraint on the upper limit value. (Note: interpret implementations must at least support the range of 6 through 72.)

5.2.2.2.21 Polyline element requirements for output. The CGM implementation for NITFS shall generate and output the Polyline element with at least two and no more than 4096 vertices.

5.2.2.2.22 Polygon and Polygon Set element requirements for output. The CGM implementation for NITFS shall generate and output the Polygon element with at least three vertices and no more than 4096 vertices.

5.2.2.2.23 Generate and output sequential order. The CGM implementation for NITFS shall generate and output CGM elements in the sequential order intended for interpretation.

5.2.2.2.24 Output Text element. The CGM implementation for NITFS shall generate and output the Text element using the following attributes. The following attributes must be stated before the first use of the Text element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Text Color

b. Character Height

c. Text Font Index

d. Character Orientation

5.2.2.2.25 Output Polygon and Polygon Set element. The CGM implementation for NITFS shall generate and output the Polygon element using the following attributes. The following attributes must be stated before the first use of the Polygon element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.2.26 Output Ellipse element. The CGM implementation for NITFS shall generate and output the Ellipse element using the following attributes. The following attributes must be stated before the first use of the Ellipse element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.2.27 Output Polyline element. The CGM implementation for NITFS shall generate and output the Polyline element using the following attributes. The following attributes must be stated before the first use of the Polyline element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Line Width

b. Line Type

c. Line Color

5.2.2.2.28 Output Elliptical Arc element. The CGM implementation for NITFS shall generate and output the Elliptical Arc element using the following attributes. The following attributes must be stated before the first use of the Elliptical Arc element but need not be restated for subsequent occurrence unless a different attribute value is required.

a. Line Width

b. Line Type

c. Line Color

5.2.2.2.29 Output Elliptical Arc Close element. The CGM implementation for NITFS shall generate and output the Elliptical Arc Close element using the following attributes. The following attributes must be stated before the first use of the Elliptical Arc Close element but need not be restated for subsequent occurrence unless a different attribute value is required.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.2.30 Output Rectangle element. The CGM implementation for NITFS shall generate and output the Rectangle element using the following attributes. The following attributes must be stated before the first use of the Rectangle element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.2.31 Output Circle element. The CGM implementation for NITFS shall generate and output the Circle element using the following attributes. The following attributes must be stated before the first use of the Circle element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.2.32 Output Circular Arc Center element. The CGM implementation for NITFS shall generate and output the Circular Arc Center element using the following attributes. The following attributes must be stated before the first use of the Circular Arc Center element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Line Width

b. Line Type

c. Line Color

5.2.2.2.33 Output Circular Arc Center Close element. The CGM implementation for NITFS shall generate and output the Circular Arc Center Close element using the following attributes. The following attributes must be stated before the first use of the Circular Arc Center Close element but need not be restated for subsequent occurrences unless a different attribute value is required.

a. Fill Color

b. Interior Style

c. Edge Visibility

d. Edge Width

e. Edge Type

f. Edge Color

5.2.2.2.34 Auxiliary Color for output. The CGM implementation for NITFS shall provide the capability to generate and output the Auxiliary Color element in support of EDGE TYPE, LINE TYPE, and TEXT.

5.2.2.2.35 Transparency for output. The CGM implementation for NITFS shall provide the capability to generate and output the Transparency element as either (1 = on) or (0 = off) in support of EDGE TYPE, LINE TYPE, and TEXT.

5.2.2.2.36 CGM element defaults for output. The CGM implementation for NITFS shall assume all CGM default values as stated in the Part 3 - Binary Encoding in the FIPS PUB 128 for each CGM that the CGM implementation generates and outputs unless otherwise specified in this document. The CGM element defaults for output include, but are not limited to, those in table CXVIII. Since this standard only specifies a single value or option, these elements, although permitted, never need to appear in a CGM.

TABLE CXVIII. CGM element defaults for output.

VDC TYPE:	16 BIT INTEGER
INTEGER PRECISION:	16 BIT INTEGER
INDEX PRECISION:	16 BIT INTEGER
COLOR PRECISION:	8 BIT INTEGER
TRANSPARENCY:	ON
TEXT PRECISION:	STRING
CHARACTER EXPANSION FACTOR:	1.0
CHARACTER SPACING:	0.0
TEXT PATH:	RIGHT
TEXT ALIGNMENT:	NORMAL HORIZONTAL, NORMAL VERTICAL
BACKGROUND COLOR:	NONE (THIS IS NITF SPECIFIC)
COLOR VALUE EXTENT:	0,0,0 - 255,255,255
VDC INTEGER PRECISION:	16 BIT INTEGER

5.2.2.2.37 Degeneracy. The CGM implementation for NITFS is precluded from generating and outputting degenerate CGM elements.

6. NOTES

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

6.1 CGM example Metafiles. The following examples illustrate complete CGM metafiles compatible with the CGM implementation for NITFS.

6.1.1 Polygon example. The following example represents a white filled polygon with three vertices at (4,5), (6,7), and (8,9) relative to the VDC Extent origin.

TABLE CXIX. Polygon example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE ("POLYGON")	0x0028 0x0750 0x4F4C 0x5947 0x4F4E
METAFILE VERSION (1)	0x1022 0x0001
METAFILE DESCRIPTION ("POLYGON")	0x1048 0x0750 0x4F4C 0x5947 0x4F4E
METAFILE ELEMENT LIST	0x1166 0x0001 0xFFFF 0x0001
CGM ELEMENT	HEX VALUES
BEGIN PICTURE ("POLYGON")	0x0068 0x0750 0x4F4C 0x5947 0x4F4E
COLOR SELECTION MODES (DIRECT = 1)	0x2042 0x0001
VDC EXTENT	0x20C8 0x0000 0x7FFF 0x7FFF 0x0000
BEGIN PICTURE BODY	0x0080
FILL COLOR (255,255,255)	0x52E3 0xFFFF 0xFF00
INTERIOR STYLE (SOLID=1)	0x52C2 0x0001
POLYGON (4,5 6,7 8,9)	0x40FF 0x000C 0x0004 0x0005 0x0006 0x0007 0x0008 0x0009
END PICTURE	0x00A0
END METAFILE	0x0040

6.1.1.1 Polygon Set example. The following example represents a polygon set which includes a red filled polygon and two smaller polygons that are transparent. The three polygons are relative to the VDC Extent origin.

TABLE CXX. Polygon Set example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE (“POLYGON SET”)	0X002C 0X0B50 0X4F4C 0X5947 0X4F4E 0X2053 0X4554
METAFILE VERSION (1)	0X1022 0X0001
METAFILE DESCRIPTION (“POLYGON SET”)	0X104C 0X0B50 0X4F4C 0X5947 0X4F4E 0X2053 0X4554
METAFILE ELEMENT LIST	0X1166 0X0001 0XFFFF 0X0001
BEGIN PICTURE (“POLYGON SET”)	0X006C 0X0B50 0X4F4C 0X5947 0X4F4E 0X2053 0X4554
COLOR SELECTION MODES (DIRECT = 1)	0X2042 0X0001
EDGE WIDTH SPECIFICATION MODE	0X20A2 0X0000
CGM ELEMENT	HEX VALUES
VDC EXTENT	0X20C8 0X0000 0X7FFF 0X7FFF 0X0000
BEGIN PICTURE BODY	0X0080
FILL COLOR (RED)	0X52E3 0XFF00 0X0000
INTERIOR STYLE (SOLID=1)	0X52C2 0X0001
EDGE VISIBILITY (ON)	0X53C2 0X0001

TABLE CXX. Polygon Set example.

EDGE WIDTH (7)	0X5382 0X0007
EDGE TYPE (DASHED)	0X5362 0X0002
EDGE COLOR (BLUE)	0X53A3 0X0000 0XFF00
CGM ELEMENT	HEX VALUES
POLYGON SET 0,0 1000,0 1000,1000, 0,1000 250,250 350,250, 350,350, 250,350 550,550 650,550 650,650 550,650	0X411F 0X0048 0X0000 0X0000 0X0001 0X03E8 0X0000 0X0001 0X03E8 0X03E8 0X0001 0X0000 0X03E8 0X0002 0X00FA 0X00FA 0X0001 0X015E 0X00FA 0X0001 0X015E 0X015E 0X0001 0X00FA 0X015E 0X0002 0X0226 0X0226 0X0001 0X028A 0X0226 0X0001 0X028A 0X028A 0X0001 0X0226 0X028A 0X0002
END PICTURE	0X00A0
END METAFILE	0X0040

6.1.2 Text example. The following CGM example represents a label with text "test" with white characters 14 pixels high starting at location (10,20) relative to VDC Extent origin. the font list consists of two fonts (Helvetica and Courier). The text uses Courier.

TABLE CXXI. Text example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE ("TEXT")	0x0025 0x0454 0x4558 0x5400
METAFILE VERSION (1)	0x1022 0x0001
METAFILE DESCRIPTION ("TEXT")	0x1045 0x0454 0x4558 0x5400
METAFILE ELEMENT LIST	0x1166 0x0001 0xFFFF 0x0001
FONT LIST ("Helvetica," "Courier")	0x11B2 0x0948 0x656C 0x7665 0x7469 0x6361 0x0743 0x6F75 0x7269 0x6572
BEGIN PICTURE ("TEXT")	0x0065 0x0454 0x4558 0x5400
COLOR SELECTION MODES (DIRECT = 1)	0x2042 0x0001
VDC EXTENT	0x20C8 0x0000 0x7FFF 0x7FFF 0x0000
BEGIN PICTURE BODY	0x0080
TEXT COLOR (255,255,255)	0x51C3 0xFFFF 0xFF00
CHARACTER HEIGHT (14)	0x51E2 0x000E
TEXT FONT INDEX (2=Courier)	0x5142 0x0002
CHARACTER ORIENTATION	0x5208 0x0000 0xFFFF 0x0001 0x0000

TABLE CXXI. Text example – Continued.

CGM ELEMENT	HEX VALUES
TEXT (10,20,1,4,"test")	0x408B 0x000A 0x0014 0x0001 0x0474 0x6573 0x7400
END PICTURE	0x00A0
END METAFILE	0x0040

6.1.3 Ellipse example. The following CGM example represents an empty ellipse with a four-pixel wide visible white edge, a center at location (10,20), with conjugate endpoints at locations (20,20) and (10,30) relative to VDC Extent origin.

TABLE CXXII. Ellipse example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE ("ELLIPSE")	0x0028 0x0745 0x4C4C 0x4950 0x5345
METAFILE VERSION (1)	0x1022 0x0001
METAFILE DESCRIPTION ("ELLIPSE")	0x1048 0x0745 0x4C4C 0x4950 0x5345
METAFILE ELEMENT LIST	0x1166 0x0001 0xFFFF 0x0001
BEGIN PICTURE ("ELLIPSE")	0x0068 0x0745 0x4C4C 0x4950 0x5345
COLOR SELECTION MODES (DIRECT = 1)	0x2042 0x0001
EDGES WIDTH SPECIFICATION MODE (ABSOLUTE=0)	0x20A2 0x0000
VDC EXTENT	0x20C8 0x0000 0x7FFF 0x7FFF 0x0000
BEGIN PICTURE BODY	0x0080
EDGE COLOR (255,255,255)	0x52A3 0xFFFF 0xFF00
EDGE VISIBILITY (on)	0x53C2 0x0001

TABLE CXXII. Ellipse example - Continued.

CGM ELEMENT	HEX VALUES
EDGE WIDTH (4)	0x5382 0x0004
EDGE TYPE (1=solid)	0x5362 0x0001
INTERIOR STYLE (EMPTY)	0x52C2 0x0004
ELLIPSE (10,20 20,20 10,30)	0x422C 0x000A 0x0014 0x0014 0x0014 0x000A 0x001E
END PICTURE	0x00A0
END METAFILE	0x0040

6.1.4 Polyline example. The following CGM example represents a three-segment open RED polyline drawn with a six-pixel wide dashed line. The vertices are at (2,2), (4,6), (3,4), and (0,6) relative to VDC Extent origin.

TABLE CXXIII. Polyline example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE ("POLYLINE")	0x0029 0x0850 0x4F4C 0x594C 0x494E 0x4500
METAFILE VERSION (1)	0x1022 0x0001
METAFILE DESCRIPTION ("POLYLINE")	0x1049 0x0850 0x4F4C 0x594C 0x494E 0x4500
METAFILE ELEMENT LIST	0x1166 0x0001 0xFFFF 0x0001
BEGIN PICTURE ("POLYLINE")	0x0069 0x0850 0x4F4C 0x594C 0x494E 0x4500
COLOR SELECTION MODE (DIRECT = 1)	0x2042 0x0001
LINE WIDTH SPECIFICATION MODE (ABSOLUTE=0)	0x2062 0x0000

TABLE CXXIII. Polyline example - Continued.

CGM ELEMENT	HEX VALUES
VDC EXTENT	0x20C8 0x0000 0x7FFF 0x7FFF 0x0000
BEGIN PICTURE BODY	0x0080
LINE COLOR (255,0,0)	0x5083 0xFF00 0x0000
LINE TYPE (DASH=2)	0x5042 0x0002
LINE WIDTH (6)	0x5062 0x0006
POLYLINE (2,2 4,6 3,4 0,6)	0x403F 0x0010 0x0002 0x0002 0x0004 0x0006 0x0003 0x0004 0x0000 0x0006
END PICTURE	0x00A0
END METAFILE	0x0040

6.1.5 Elliptical Arc example. The following CGM example represents a blue elliptical arc that is four pixels wide. Figure 6 demonstrates pictorially the Elliptical Arc parameters and the relationship between conjugate diameter endpoints and start and stop vectors.

TABLE CXXIV. Arc example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE ("ARC")	0x0024 0x0341 0x5243
METAFILE VERSION (1)	0x1022 0x0001
METAFILE DESCRIPTION ("ARC")	0x1048 0x0341 0x5243
METAFILE ELEMENT LIST	0x1166 0x0001 0xFFFF 0x0001
BEGIN PICTURE ("ARC")	0x0068 0x0341 0x5243
COLOR SELECTION MODES (DIRECT = 1)	0x2042 0x0001
VDC EXTENT	0x20C8 0x0000 0x7FFF 0x7FFF 0x0000
BEGIN PICTURE BODY	0x0080
LINE WIDTH (4)	0x5062 0x0006
LINE COLOR (0,0,255)	0x5083 0x0000 0xFF00
LINE TYPE (1=solid)	0x5042 0x0001
ELLIPTICAL ARC (10,10 10,5 20,10, 15,5 5,5)	0x4254 0x0014 0x000A 0x000A 0x000A 0x0005 0x0014 0x000A 0x000F 0x0005 0x0005 0x0005
END PICTURE	0x00A0
END METAFILE	0x0040

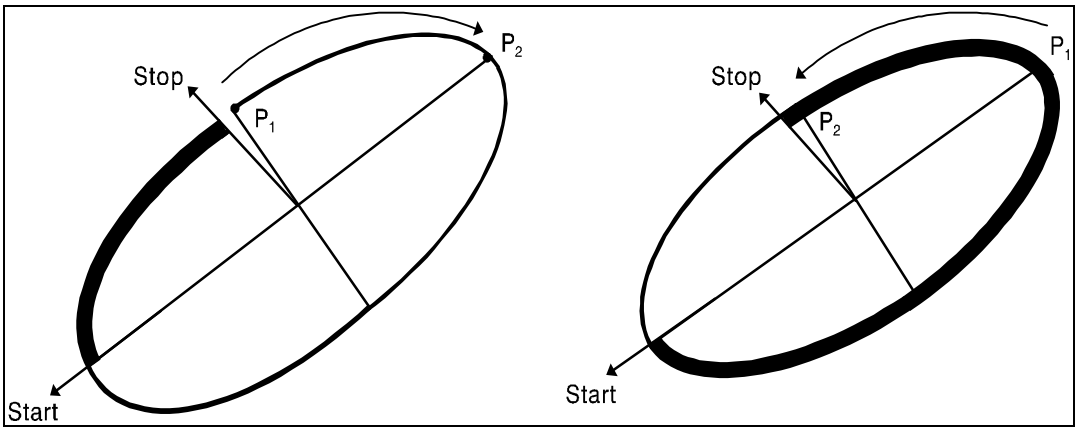

FIGURE 6. Eliptical Arc orientatopm.

6.1.6 Elliptical Arc example. The following CGM example represents an elliptical arc close that is red and filled with pie closure.

TABLE CXXV. Arc Closed example.

CGM ELEMENT	HEX VALUES
BEGIN METAFILE ("ARCCLOSE")	0x0024 0x0841 0x5243 0x434C 0x4F53 0x4500
METAFILE VERSION (1)	0x1022 0x0001
METAFILE DESCRIPTION ("ARC")	0x1048 0x0841 0x5243 0x434C 0x4F53 0x4500
METAFILE ELEMENT LIST	0x1166 0x0001 0xFFFF 0x0001
BEGIN PICTURE ("ARC")	0x0068 0x0841 0x5243 0x434C 0x4F53 0x4500
COLOR SELECTION MODE (DIRECT = 1)	0x2042 0x0001
VDC EXTENT	0x20C8 0x0000 0x7FFF 0x7FFF 0x0000
BEGIN PICTURE BODY	0x0080

TABLE CXXV. Arc Closed example – Continued.

CGM ELEMENT	HEX VALUES
INTERIOR STYLE	0x5062 0x0006
FILL COLOR (255,0,0)	0x5083 0x0000 0xFF00
EDGE VISIBILITY (on)	0x53C2 0x0001
ELLIPTICAL ARC CLOSE (10,10 10,5 20,10, 15,5 5,5)	0x4276 0x000A 0x000A 0x000A 0x0005 0x0014 0x000A 0x000F 0x0005 0x0005 0x0005 0x0000
END PICTURE	0x00A0
END METAFILE	0x0040

6.2 Color to gray scale conversion. Full color may be specified for Attribute Elements dealing with color. Color items for receiving systems unable to support full color must be mapped to colors they are able to support.

For eight-bit gray scale systems:

$$\text{pixel value}_8 = 0.299 * \text{RED} + 0.587 * \text{GREEN} + 0.114 * \text{BLUE}$$

For one bit black and white systems:

$$\text{pixel value}_1 = 1 \text{ (white), if pixel value}_8 > 127$$

$$0 \text{ (black), if pixel value}_8 \leq 127$$

6.3 Subject term (key word) listing.

Image
Picture
Secondary Imagery Dissemination System
SIDS
Symbol(s)

6.4 Changes from previous issue. Marginal notations are not used in this revision to identify changes with respect to the previous issue due to the extent of the changes. The following is a list of features that have been added to the NITFS CGM implementation as described in this document.

a. New Metafile Description

This revised standard defines a new metafile descriptor (NITF/CGM;ProfileED:2301-2/Source:*producer*;Date:YYYYMMDD) to align it with the international standard upon which this standard is based. For legacy purposes, the previous metafile (NITF/CGM-APP-2.0) may

continue to be used when the content of the metafile does not contain features which have been added or modified by this revised standard. NITF 2.0 formatted files will not use the new descriptor nor the modified/add features of this standard. NITF 2.1 formatted files should use the new descriptor and may exercise the new/modified features. However, the old descriptor may also appear in NITF 2.1 files for example when re-using existing CGM's produced under the previous version of this standard.

b. Changed features

Edge Visibility of (0=off, 1=on) vice)1=on).

Edge Width of 1 through 100 vice 2, 4, 6.

Line Width of 1 through 100 vice 2, 4, 6.

Edge Types of Solid, Dashed, Dotted, Dot-dashed and Dot-dot-dashed vice Solid and Dashed.

Line Types of Solid, Dashed, Dotted, Dot-dashed and Dot-dot-dashed vice Solid and Dashed.

Font Sizes of 6 through 72 vice 10 through 38.

Interior Styles of Solid, Hatch and Empty vice Solid and Empty.

Transparency of (0=off, 1=on) vice (1=on).

Character Set Basic Latin (0020-007E), Latin-1 Supplement (00A0-00FF) and null 0x00 vice Basic Latin (0020-007E).

Default Elements may optionally be included vice not including.

c. Additional features

Auxiliary Color

Polygon sets

CONCLUDING MATERIAL

Custodians:
Army - CR
Navy - OM
Air Force - 02
Other - NS

Review activities:
OASD - DO, IR, SE
Army – AC, AC1, AC2, CR5, PT, TM1, TM3
Navy – AS, CG, CH, EC, MC, NC, ND, TD
Air Force - 11, 13, 16, 17, 19, 29, 33, 90, 93
DLA - DH
Misc – DC3, DC4, DI, US

Civil agency coordinating activities:
USDA – AFS, APS
DOC - NIST
DOE
EPA
GPO
HHS – NIH
DOI – BLM, GES, MIN
DOT – CGCT, OST

Preparing activity:
Misc - MP

Agent:
Not applicable

(Project IPSC-0500)

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-2301A**	2. DOCUMENT DATE (YYMMDD) **980605**

3. DOCUMENT TITLE
Computer Graphics Metafile (CGM) Implementation Standard for the National Imagery Transmission Format Standard

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
National Imagery and Mapping Agency

a. NAME **Danny Rajan**	b. TELEPHONE *(Include Area Code)* (1) Commercial **(703) 262-4416** (2) AUTOVON
c. ADDRESS *(Include Zip Code)* **MS P-24** **12310 Sunrise Valley Drive** **Reston, VA 20191-3449**	**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