NOTICE OF CHANGE **NOT MEASUREMENT SENSITIVE**

MIL-STD-2301A
NOTICE 1
1 March 2001

DEPARTMENT OF DEFENSE
INTERFACE STANDARD

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

TO ALL HOLDERS OF MIL-STD-2301A:

1. THE FOLLOWING PAGES OF MIL-STD-2301A HAVE BEEN REVISED AND SUPERSEDE THE PAGES LISTED:

NEW PAGE	DATE	SUPERSEDED PAGE	DATE
Cover	5 June 1998	Cover	reprinted without change
ii	5 June 1998	ii	1 March 2001
5	5 June 1998	5	1 March 2001
6	5 June 1998	6	1 March 2001
15	5 June 1998	15	1 March 2001
16	5 June 1998	16	1 March 2001
18	5 June 1998	18	1 March 2001
19	5 June 1998	19	reprinted without change
20	5 June 1998	20	1 March 2001
23	5 June 1998	23	1 March 2001
32	5 June 1998	32	1 March 2001
DD 1426	14 March 1997	DD 1426	1 March 2001

2. RETAIN THIS NOTICE AND INSERT BEFORE TABLE OF CONTENTS.

3. Holders of MIL-STD-2301A will verify that the page changes and additions indicated above have been entered. This notice page will be retained as a check sheet. This issuance, together with appended pages, is a separate publication. Each notice is to be retained by stocking points until the military standard is completely revised or canceled.

Custodians:
Army – CR
Navy – OM
Air force –02
Other – NS

Preparing Activity:
NIMA - MP

NOT MEASUREMENT SENSITIVE

MIL-STD-2301A
5 June 1998
SUPERSEDING
ON 1 March 2001
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**

REPRINTED WITHOUT CHANGE

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), 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.

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)
HATCH INDEX
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 15 14 13 12	11 10 9 8 7 6 5	4 3 2 1 0 LSB
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]
[HATCH INDEX]
[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]

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					= 0x53C2
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					= 0x5062
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 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(3). Note, the parameter list length is

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 0 LSB	
4	17	12 = 0x422C
CenterX		
CenterY		
End1X		
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 0 LSB	
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 0 LSB	
4	1	31 = 0x403F
Parameter list length (4N)		
Vertex1X		
Vertex1Y		
Vertex2X		
Vertex2Y		
:		
Vertex(N)X		
Vertex(N)Y		

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 0 LSB	
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 0 LSB	
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. The Red Green, and Blue (RGB) values are specified

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					= 0x53C2
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.

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 **National Imagery and Mapping Agency**	b. TELEPHONE *(Include Area Code)* (1) Commercial (**703) 262-4416** (2) AUTOVON
c. ADDRESS *(Include Zip Code)* **12310 Sunrise Valley Drive** **Reston II Building, Mailstop P24** **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