NISTIR 4579

NIST Support for the Computer-Aided Acquisition and Logistic Support (CALS) Program in the Area of Graphics Standards, Calendar Year 1990

Daniel R. Benigni
Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Computer Systems Laboratory
Gaithersburg, MD 20899

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 4579

NIST Support for the Computer-Aided Acquisition and Logistic Support (CALS) Program in the Area of Graphics Standards, Calendar Year 1990

Daniel R. Benigni
Editor

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Computer Systems Laboratory
Gaithersburg, MD 20899

May 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

ensures 100 % accurate exchange of graphical data metafiles between between DoD and contractors. NIST recognized early on that having the CGM standard in place, even if it contained all the functionality necessary to meet CALS Program requirements, would not be adequate to ensure 100 % accurate exchange of graphical data metafiles. Requirements concerning the behavior of either metafile generators or interpreters are not part of the CGM standard, causing variations between implementations of these components. The CGM Application profile deals with these issues, and provides the means for satisfying A. above until the CGM standard is revised. The NIST Graphics Software Group developed the CGM Application Profile for CALS in the form of a Military Specification called MIL-D-28003, first published in December 1988. At present, Revision A of MIL-D-28003 is being prepared for DoD and industry review. Future revisions will probably occur at two-year intervals.

C. **Develop the software tools necessary to assure conformance to both the CGM standard and MIL-D-28003.** When the NIST Graphics Software Group first began work on CALS, there existed no test routines for the CGM standard. They had to be created, beginning with functional specifications and a conceptual design. Then test requirements had to be documented. Finally, software that tests conformance both to the CGM standard and MIL-D-28003 had to be written. The NIST Graphics Software Group has completed this work, and is now beta-testing this software test tool. Plans are underway to extend testing next to metafile generators, and finally to metafile interpreters.

This report comprises the deliverables (which have not necessarily been endorsed by the CALS Office) of the NIST Graphics Software Group in support of the CALS Program for FY90, continuing the efforts described above. The separate task deliverables assigned were as follows:

1. **Update CGM Application Profile (i.e., prepare MIL-D-28003A);**

2. **Inject CALS requirements into the standards committees working on the Amendments to the CGM standard;**

3. **Produce a software tool to determine conformance of a metafile to the CGM standard and to MIL-D-28003; and**

4. **Explore potential sources of generator/interpreter conformance test capabilities.**

PREVIOUS NIST GRAPHICS SOFTWARE GROUP WORK FOR CALS

The work of the NIST Graphics Software Group in support of the CALS Program from FY86 to present can be found in the following NIST Internal Reports (NISTIRs, formerly NBSIRs):

Kemmerer, S., Editor, "Final NBS Report for CALS, FY86," U.S. Department of Commerce, National Bureau of Standards, NBSIR 87-3566, May 1987.

Kemmerer, S., Editor, "A Collection of Technical Studies Completed for the Computer-aided and Acquisition and Logistic Support (CALS) Program, Fiscal Year 1987," U.S. Department of Commerce, National Bureau of Standards, NBSIR 88-3727, March 1988.

Morgan, Roy S., Editor, "A Collection of Technical Studies Completed for the Computer-aided and Acquisition and Logistic Support (CALS) Program, Fiscal Year 1988," U.S. Department of Commerce, National Institute of Standards and Technology, NISTIRs 4315, 4316, and 4317, April 1990.

Kemmerer, Sharon J. and Skall, Mark W., "Graphics Application Programmer's Interface Standards and CALS," U.S. Department of Commerce, National Institute of Standards and Technology, NISTIR 89-4199, October 1989.

Benigni, Daniel R., "Graphics Standards in the Computer-aided and Acquisition and Logistic Support (CALS) Program, Fiscal Year 1989, Volume 1: Test Requirements Document and Extended CGM (CGEM)," U.S. Department of Commerce, National Institute of Standards and Technology, NISTIR 4329, May 1990.

Benigni, Daniel R., "Graphics Standards in the Computer-aided and Acquisition and Logistic Support (CALS) Program, Fiscal Year 1989, Volume 2: MIL-D-28003 Revisions and CGM Registration," U.S. Department of Commerce, National Institute of Standards and Technology, NISTIR 4330, May 1990.

CONTRIBUTORS

NIST would like to acknowledge the major technical contributors to the separate reports contained herein. They are:

> Henderson Software Company, in particular Mr. Lofton Henderson, for his work on updating the CGM Application Profile and injecting CALS requirements at standards meetings;
>
> CGM Technology Software, Inc., for work on the software conformance tool; and
>
> Peter R. Bono Associates, Inc., in particular Dr. Peter R. Bono, for his work on sources of conformance test capabilities.

The editor would also like to gratefully acknowledge the efforts of those who participated in the review process of the documents presented in this report, namely:

David K. Jefferson
Sharon J. Kemmerer
Lynne Rosenthal
Mark W. Skall

FINAL REPORT

CALS CY90 SOW TASK 4.1.1

MIL-D-28003 REVISION A RECOMMENDATIONS
AND
DRAFT TEXT OF MIL-D-28003A

Update MIL-D-28003

PURPOSE AND BACKGROUND

This report describes activities during CY90 concerned with increasing CGM applicability in CALS by **updating the CGM Application Profile (AP), or MIL-D-28003** (CALS SOW Task 4.1.1).

This report provides draft text for MIL-D-28003 Revision A, MIL-D-28003A for short, as of December 1990. This report also recommends a future course of action for CY91, specifically aimed at having Revision A of MIL-D-28003 completed in time for the CALS EXPO in November of 1991.

This task concerned two major subtasks: (a) reconciliation of the CALS and TOP APs of CGM, and (b) recommendations for modifications and extensions of the CALS AP leading to the production of draft text for MIL-D-28003A.

In 1989 CALS and MAP/TOP technical personnel reached substantial technical agreement on unifying their respective profiles. MAP/TOP sent a proposal for coordinated development to the CALS Office. Administrative action was required by both sides to create and execute the mechanisms for consolidating the projects. In lieu of any action the CALS profile has diverged from the MAP/TOP profile because of substantial improvements being made to the CALS profile through this revision process.

SUMMARY AND RECOMMENDATIONS

Due to problems in the standards process for the Extended CGM standard, or CGEM, a number of options had to be addressed and resolved concerning revision work for MIL-D-28003 during this calendar year. These options were as follows:

A. Stick to the plan as described in the 1989 Final Report and forge ahead, with full changes as recommended in that report.

B. Postpone processing of MIL-D-28003A altogether; possibly publish a technical report promoting and explaining how to use MIL-D-28003 effectively, and soliciting industry feedback.

C. (A middle course) Publish a MIL-D-28003A scaled back from the recommendations of the 1989 Final Report but addressing a few critical areas known to be causing interchange problems. There are two variations on this middle course:

(1) Include anything not in conflict with standard extensions to the CGM standard in progress, particularly CGM Amendment 3. This would allow some of the more serious problems-those involving text and fonts-to be addressed.

(2) Include the most critical extensions of CGM Amendment 3 in a draft MIL-D-28003A that would commence review early in CY91, and that would finish review shortly after the anticipated completion of CGM Amendment 3.

Option C(2) was selected. Given the timing of MIL-D-28003A review and Amendment 3 processing, it was imprudent to adopt the registered versions (i.e., those being processed as part of the Register of Graphical Items) of the needed Amendment 3 functions. It was also imprudent to wait two or three years for MIL-D-28003 Revision B to include these Amendment 3 functions.

Therefore, a certain number of the most useful Amendment 3 functions will be adopted in the draft of MIL-D-28003A to be reviewed early in 1991. The addition of these functions will strike a balance between utility in technical illustration on the one hand and burden to implement on the other. Anticipated for inclusion are most of the line and drawing controls, hatch definitions, restricted text controls, compressed raster primitives, and some additional curve definitions. In addition it would be very useful if some symbol set capabilities, even if only those from the original Hershey work, could be included. Finally, an unambiguous mapping of 7-bit and 8-bit codes to glyphs must be prepared for all of the fonts allowed in MIL-D-28003A.

This final report contains a draft MIL-D-28003A that still requires additional work in the areas described above before the CALS industry and DoD review process can occur in early 1991.

Recommendation: NIST should complete the above described work early in 1991. Then DoD and Industry review can commence so that MIL-D-28003A can be completed and published prior to CALS Expo in early November of 1991.

Update MIL-D-28003

Recommendation: NIST should prepare an appendix to the draft MIL-D-28003A to circulate with the draft, describing changes from MIL-D-28003 to MIL-D-28003A and rationale for these changes.

Recommendation: NIST should prpeare an appendix to the draft MIL-D-28003A to circulate with the draft, previewing a MIL-D-28003B containing all of the extensions proposed and recommended by NIST in previous reports.

REFERENCES

The draft of MIL-D-28003A that follows refers to the following key documents:

> **CGM Amendment 1:** the first set of formal standard extensions of CGM; final text is available; final administrative action was taken by the ISO Central Secretariat in Geneva, and the amendment is part of CGM as of 1 November 1990.
>
> **Registration Proposals:** exist in various X3H3 documents; are contained in ISO Register of Graphical Items. (Application for copies should be addressed to the ISO Registration Authority, National Institute of Standards and Technology, Building 225, Room A266, Gaithersburg, MD 20899.)
>
> **CGM Amendment 3:** Draft Amendment text dated November 1990 and currently awaiting DAM ballot commencement in SC24/WG3; anticipated completion is August 1991.

CURRENT TEXT OF DRAFT MIL-D-28003A

Appendix 1 contains the draft text of MIL-D-28003A as of the end of CY90. Areas of text and tables referred to as either TBD or **x.y.z** will be completed before submission for CALS review in early CY91. That means that this draft version will undergo significant change prior to DoD and industry review during 1991.

APPENDIX 1

DRAFT OF MIL-D-28003A

NON-MEASUREMENT SENSITIVE

MIL-D-28003A
31 December 1990

MILITARY SPECIFICATION

DIGITAL REPRESENTATION FOR COMMUNICATION OF ILLUSTRATION DATA: CGM APPLICATION PROFILE

This specification is approved for use by all Departments and Agencies of the Department of Defense.

1. SCOPE

1.1 Scope. This military specification establishes the requirements to be met when 2-dimensional picture description or illustration data that is vector or mixed vector and raster is delivered in the digital format of the Computer Graphics Metafile (CGM) as specified by its Federal Information Processing Standard, FIPS PUB 128.

Beneficial comments (recommendations, additions, deletions) and any pertinent data which may be used in improving this document shall be addressed to: Director, CALS Policy Office, DASD(S)CALS Pentagon, Room 2B322, Washington, DC 20301, by using the self addressed Standardization Document approval Proposal (DD Form 1426) appearing at the end of this document or by letter.

AMSC N/A AREA ILSS

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

1.2 <u>Classification</u>. This specification establishes the requirements for the communication or interchange of illustration data in digital format for use in technical illustrations and publications. The CGM Application Profile (AP) defined by this specification consists of three parts: the metafile, the generator, and the interpreter. There shall be only one level for the generator, and it shall be called conforming basic generator. Metafiles shall be conforming according to one of these categories:

--monochrome (black & white);

--grayscale

--full color.

Interpreters shall be classified and conform similarly to metafiles, according to one of the three categories listed above.

(Editors note: the grayscale needs to be considered in terms of the basic set for conforming metafiles and minimum capabilities for interpreters -- 16 levels, or 32, or 256, or what?)

2. APPLICABLE DOCUMENTS

2.1 Government documents.

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

STANDARDS

FEDERAL

FIPS PUB 128 - Computer Graphics Metafile (CGM)

Note: FIPS PUB 128 adopts ANSI/ISO 8632 and as a Federal Information Processing Standard Publication (FIPS PUB).

(Copies of the referenced Federal Information Processing Standards are available to Department of Defense activities from the Commanding Officer, Naval Publications and Forms Center, 5801 Tabor Avenue, Philadelphia, PA 19120-5099. Others must request copies of FIPS from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.)

MILITARY

MIL-STD-1840A - Automated Interchange of Technical Information

(Copies of the referenced military standard are available from the Department of Defense Single Stock Point, Commanding Officer, Naval Publications and Forms Center, 5801 Tabor Avenue, Philadelphia, PA 19120.)

2.1.2 Other Government documents. The following other Government document forms a part of this document to the extent specified herein. Unless otherwise specified, the issue is that cited in the solicitation.

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

NBS SP 424 - A Contribution to Computer Typesetting Techniques: Tables of Coordinates for Hershey's Repertory of Oxidental Type Fonts and Graphic Symbols, NBS Special Publication 424, April 1976.

(Application for copies shall be addressed to the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.)

2.2 Non-Government publications. The following documents 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 (see 6.2).

INTERNATIONAL STANDARDS ORGANIZATION

ISO 8632-1:1987/Amendment 1:1990 Computer Graphics Metafile (CGM)

ISO 8632-1:1987/DAM 3, Amendment 3:1991 Computer Graphics Metafile (CGM)

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

ISO Register of Graphical Items

(Application for copies shall be addressed to the ISO Registration Authority, National Institute of Standards and Technology, Building 225, Room A266, Gaithersburg, MD 20899.)

NATIONAL STANDARDS

ANSI X3.4 - 7-bit American National Standard Code for Information Interchange (7-bit ASCII)

ANSI X3.134/2 - 8-bit American National Standards Code for Information Interchange (8-bit ASCII)

(Application for copies shall be addressed to: American National Standards Institute, Inc., 1430 Broadway, New York, NY 10018).

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

2.3 Order of precedence. In the event of a conflict between the text of this specification and the references cited herein, the text of this specification shall take precedence. Nothing in this specification, however, shall supersede applicable laws and regulations unless a specific exemption has been obtained.

3. REQUIREMENTS

3.1 General requirements. This specification defines conformance of a CGM metafile in terms of "permissible" and "basic" values. Permissible values are the range of values of CGM elements as specified in FIPS PUB 128. Basic values are a subset of the permissible values, in some cases augmented by additional values contained in this specification, and they constitute the "Basic Set." For example, permissible values of MARKER TYPE include all non-zero integers, while basic values are limited to the specific values 1 to 5. A conforming basic metafile shall contain no elements or parameters outside of the Basic Set. The CGM AP which corresponds to the illustration data to be communicated shall be in the form of one or more conforming basic metafiles.

3.1.1 Conforming basic generator. A conforming basic generator shall be defined to be one that produces only conforming basic metafiles (or can be reliably commanded to function in that mode), and additionally conforms to any additional generator requirements as explained in the subsections below.

3.1.2 Conforming basic interpreter. A conforming basic interpreter shall be defined to be one that at least correctly interprets any conforming basic metafile, and conforms to any additional interpreter requirements as explained in the subsections below. A conforming basic interpreter can conform to any of the three categories defined in section 1.2 above: full color, grayscale, or monochrome. In addition, any conforming basic interpreter shall be able to parse and skip any elements that it does not understand or support, and any parameter values that it does not support. For all conforming interpreters: all of the specifications of this CGM AP shall be accurately implemented. All of the specifications of FIPS PUB 128 shall be accurately implemented except as modified by this specification. This includes the guidelines of FIPS PUB 128 annex D.2 and D.5, and the recommendations for the treatment of indeterminate specifications of circular and elliptical primitives in FIPS PUB 128 annex D.4.5. The results shall be completely predictable across implementations conforming at this level; that is, suitable for publication.

Conforming basic interpreters shall render all text at "stroke" precision, regardless of the value of the metafile TEXT PRECISION element.

(Editors note: should the following even be in 28003A, given the removal of Draft Level?)

In the case that the output device has only fewer colors available or not exactly the same set of colors as specified in the metafile, metafile colors shall be mapped to device colors as follows. If the metafile color selection mode is "direct" the value of the metafile BACKGROUND COLOUR shall map to one of the device colors (the background color); any color value of any other metafile element which is exactly equal to the value of the metafile BACKGROUND COLOUR shall also map to the device background color; all other color values in the metafile shall map to another device color, which must be distinct from the device background color, and which must be closest to the specified metafile color according to some reasonable metric applied to color space. If the metafile color selection mode is "indexed", only the BACKGROUND COLOUR and COLOUR TABLE elements contain RGB values to be mapped. The metafile "effective background color" is defined to be the value of the BACKGROUND COLOUR element, or the value of the COLOUR TABLE setting of index 0 if the BACKGROUND COLOUR has been thus superseded. Then the effective background color shall map to one of the device colors (the background color); any COLOUR TABLE values which exactly match the effective background color shall also map to this value; all other RGB values shall map to another device color, which must be distinct from the device background color, and which must be closest to the specified metafile color according to some reasonable metric applied to color space.

3.1.3 Limits on parameter data. A conforming basic metafile shall not contain scalar values of parameter data outside the ranges specified by this specification.

3.1.4 Encoding format. A conforming basic metafile shall use only the CGM Binary Encoding, as defined in FIPS PUB 128, part 3.

3.1.5 Physical file structure. This specification does not define media delivery requirements. These should be specified by the standards concerned with integration of digital document content types (e.g., MIL-STD-1840A or MIL-HDBK-59) or delivery of digital technical information. In the absense of such guidance it is recommended that metafiles conforming to this specification be delivered as continuous byte streams where the media and environment permit (e.g., diskette or similar media on "PC" systems).

3.1.6 Errors in FIPS PUB 128. A number of editorial errors have been found to exist in the published version of ANSI X3.122. In order to prevent errors in the use of FIPS PUB 128 within this specification, the following changes to ANSI X3.122 shall apply:

Part 1, p. 100, the last item on the page: "1" should be "0" and "foreground" should be "background".

Part 3, p.17, item 11: the fraction numerator which is "pnx" should be "pnx-1".

Part 3, p.26, VDC REAL PRECISION: "3I" should be "E,2I".

Part 1, clause 5.2.1 (p. 43), clause 5.3.12 (p. 49), and clause 6 (p. 100): To make clear and remove contradictory statements in these clauses--Metafile Descriptor elements shall not return to default at BEGIN PICTURE, and they shall not be included in the METAFILE DEFAULTS REPLACEMENT.

Part 1, p.106, the expansion of "<metafile contents>": the "¦" symbols should be deleted.

(Editors note: this should be updated before publication of 28003A with the output of the ISO CGM Metafile Maintenance Rapporteur Group, which output is anticipated to be available by the time the review of this amendment is published).

3.2 Specific requirements. The following subsections define the specific requirements for conforming metafiles, generators, and interpreters. An application profile shall use the specified element types of FIPS PUB 128 with the constraints as specified below.

3.2.1 Metafile constraints. The Basic Set shall be defined by the limitations on Basic Values noted below. Where an element is not mentioned, it is implied that the Basic Set shall include all values permitted in FIPS PUB 128.

3.2.1.1 Delimiter elements. There are no parameter range contraints imposed upon any of the delimiter elements, other than the string length limits, Name Precision limits, and segment count limits defined in later sections of this specification.

3.2.1.2 Metafile descriptor elements. The metafile descriptor element constraints shall be as specified in table I.

TABLE I. Metafile descriptor element constraints

Element	Basic Values
METAFILE VERSION	1,2 (Note 1)
METAFILE DESCRIPTION	(Note 2)
INTEGER PRECISION	16
REAL PRECISION	(1,16,16) (fixed point) (0,9,23) (floating point)
INDEX PRECISION	16
COLOUR PRECISION	8, 16
COLOUR INDEX PRECISION	8, 16
FONT LIST	(Note 3)
CHARACTER SET LIST	(0,4/2) (Note 4) (1,4/1) (Note 5)
CHARACTER CODING ANNOUNCER	0 (Basic 7-bit) 1 (Basic 8-bit)
MAXIMUM COLOUR INDEX	0..255 (Note 6)
Elements from CGM Amendment 1:1990	
NAME PRECISION	8,16
MAXIMUM VDC EXTENT	no restrictions
SEGMENT PRIORITY EXTENT	no restrictions
Elements from CGM Amendment 3:1991	
TBD.	

Note 1: Both the version value "1" of the base CGM standard, ANSI/ISO 8632:1987, and the version value "2" of "CGM amendment 1", are basic values.

Note 2: There shall be exactly one METAFILE DESCRIPTION element in each metafile. The METAFILE DESCRIPTION element's string: a) shall include a substring briefly identifying the generator of this metafile, including company, product, and product version; b) shall contain the substring "MIL-D-28003A/BASIC-1"; c) shall have appended to this latter string either:

--".0" if the metafile is monochrome;
--".1" if the metafile is grayscale;
--".2" or "" (nothing, a null string) if the metafile is color.

Note 3: Thirty-two simultaneous fonts are supported. The font names are selected from the basic font names in 3.2.5.

Note 4: The character set is ANSI X3.4, 7-bit American National Standard Code for Information Interchange (7-bit ASCII).

Note 5: The character set is ANSI X3.134/2, 8-bit American National Standards Code for Information Interchange (8-bit ASCII). [Note: This is equivalent to ISO 8859/1, Right-Hand Part of Latin Alphabet Number 1.]

Note 6: MAXIMUM COLOUR INDEX applies to all color indexes defined or otherwise referenced, whether they are referenced implicitly or explicitly. This should be clear in FIPS PUB 128 but is not. An example of an implicit reference is a COLOUR TABLE element which defines 100 entries starting at index 250. Only index 250 is explicitly referenced by this element, but index 307, for example, is defined and therefore implicitly referenced.

3.2.1.3 Picture descriptor elements. Picture descriptor element constraints shall be as specified in table IA.

TABLE IA. Picture descriptor element constraints

Element	Basic Values
COLOUR SELECTION MODE	(Note 1)
SCALING MODE	(Note 2)
Elements from CGM Amendment 1:1990	
SET LINE REPRESENTATION	element not allowed
SET MARKER REPRESENTATION	element not allowed
SET TEXT REPRESENTATION	element not allowed
SET FILL REPRESENTATION	element not allowed
SET EDGE REPRESENTATION	element not allowed
DEVICE VIEWPORT	element not allowed
DEVICE VIEWPORT MAPPING	element not allowed
DEVICE VIEWPORT SPECIFICATION MODE	element not allowed
Elements from CGM Amendment 3:1991	
TBD.	

Note 1: Only a single value of COLOUR SELECTION MODE shall be allowed in each picture.

Note 2: The scale-factor parameter of SCALING MODE is always a floating point number, even when REAL PRECISION has selected fixed point for other real numbers. It is not apparent in FIPS PUB 128 what the precision of this floating point parameter is when fixed point reals have been selected: its precision shall be (0,9,23).

3.2.1.4 Control elements. Control element constraints shall be as specified in table II.

TABLE II. Control element constraints

Element	Basic Values
VDC INTEGER PRECISION	16, 32
VDC REAL PRECISION	(1,16,16) (fixed) (0,9,23) (floating point)
TRANSPARENCY	1 (on)
Elements from CGM Amendment 1:1990	
LINE CLIP MODE	1 (shape) (Note 1)
MARKER CLIP NODE	1 (shape) (Note 2)
EDGE CLIP MODE	1 (shape) (Note 1)
NEW REGION	no restrictions
SAVE PRIMITIVE CONTEXT	element not allowed
RESTORE PRIMITIVE CONTEXT	element not allowed
Elements from CGM Amendment 3:1991	
TBD.	

Note 1: Because the single allowed value is not the default value, this element must appear in every metafile conforming to this specification, either in the picture body or in a Metafile Defaults Replacement.

3.2.1.5 Graphical primitives. To ensure portability and predictability of results, conforming basic metafiles shall not contain any Generalized Drawing Primitive (GDP) elements. [**Note:** Future addenda to this specification may specify GDP elements to be included in the Basic set.]

The constraints in Table IIA shall apply to the parameter lists of graphical primitive elements. In the table "npts" refers to the number of points in a point list.

TABLE IIA. Graphical primitive constraints

Element	Constraints
POLYLINE	npts=2,3,4..1024
POLYMARKER	npts=1,2,3..1024
DISJOINT POLYLINE	npts=2,4,6..1024
POLYGON	npts=3,4,5..1024
POLYGON SET	npts=3,4,5..1024 (Note 1)
TEXT	(Note 2)
APPEND TEXT	(Note 2)
RESTRICTED TEXT	(Note 2)
Elements from CGM Amendment 1:1990	
Closed Figure	(Note 3)
CONNECTING EDGE	no constraints
CIRCULAR ARC CENTER REVERSED	no constraints
Elements from CGM Amendment 3:1991	
TBD.	

Note 1: In addition, any sub-polygon must be well defined and have at least 3 points.

Note 2: The string parameters of graphical text shall not contain any control characters (codes in the ranges 0..31 and 97..127) except as allowed by and necessary for the character set switching modes of CHARACTER CODING ANNOUNCER which are allowed in this specification).

Note 3: Constraints on this element include constraints defined elsewhere in this specification on the individual components of this element (e.g., polyline vertex constraints). Other constraints, on the entire assembly comprising this element, are TBD.

3.2.1.6 Attribute elements. Attribute element constraints shall be as specified in table III.

TABLE III. Attribute element constraints

Element	Basic Values
LINE BUNDLE INDEX	1-5
LINE TYPE	1-5, 6-15 (Note 1)
LINE WIDTH	positive
MARKER BUNDLE INDEX	1-5
MARKER TYPE	1-5
MARKER SIZE	positive
TEXT BUNDLE INDEX	1-2
TEXT FONT INDEX	1-32 (Note 2,3)
CHARACTER HEIGHT	positive
CHARACTER SET INDEX	1-2 (Note 2,4)
ALTERNATE CHARACTER SET INDEX	1-2 (Note 2,4)
FILL BUNDLE INDEX	1-5
HATCH INDEX	1-6
EDGE BUNDLE INDEX	1-5
EDGE TYPE	1-5
PATTERN TABLE	Starting Index, 1-8 nx, 1-16 ny, 1-16
COLOUR TABLE	start index 0-255
Elements from CGM Amendment 1:1990	
PICK IDENTIFIER	(Note 5)
Elements from CGM Amendment 3:1991	
TBD.	

Note 1: The line types defined in 3.2.2.1 shall be included in the Basic Set of this specification, and comprise the registered index values 6-15.

Note 2: The character set selected shall be representable in the font selected (see discussion of resolution of conflicts with this requirement in later sections).

Note 3: Additionally, every referenced font index shall correspond to a defined entry in the FONT LIST.

Note 4: Additionally, every referenced character set index shall correspond to a defined entry in the CHARACTER SET LIST.

Note 5: This element has no graphical effect and may be useful for preserving non-graphical application information. Therefore it is harmless when occurring in a metafile and may safely be ignored by interpreters.

For indexed color selection, either the background color and all color indexes used in the metafile shall have their representations defined or none shall. Color indexes are defined by the COLOUR TABLE element. Background color is defined either by the BACKGROUND COLOUR element or by the definition of color index 0 (BACKGROUND COLOUR is synonymous with color index 0 \(em this is part of FIPS PUB 128 but is not apparent in the original text). A color index is "used" if it occurs in an element selecting a color value to be applied to a primitive (LINE COLOR, CELL ARRAY, etc). A color index is also "used" if it is the default for a primitive attribute and the default applies to a displayed primitive. The background colour is automatically "used".

For direct color selection, either the background color and the color of each displayed primitive shall be explicitly defined, or none shall. In other words, either all colors shall be defaulted or none shall.

3.2.1.7 Segment elements. Segment element constraints shall be as specified in table IIIA.

TABLE IIIA. Segment element constraints

Element	Basic Values
Elements from CGM Amendment 1:1990	
COPY SEGMENT	no restrictions
INHERITANCE FILTER	no restrictions
CLIP INHERITANCE	element not allowed
SEGMENT TRANSFORMATION	no restrictions
SEGMENT HIGHLIGHTING	element not allowed
SEGMENT DISPLAY PRIORITY	no restrictions
SEGMENT PICK PRIORITY	no restrictions (Note 1)
Elements from CGM Amendment 3:1991	
TBD.	

Note 1: This element has no graphical effect and may be useful for preserving application information. Therefore it is harmless when occurring in a metafile and may safely be ignored by interpreters.

Both global segments and local segments are allowed in conforming basic metafiles. When global segments are defined, in the Metafile Descriptor, all global segment definitions must follow all other Metafile Descriptor elements.

3.2.1.7 ESCAPE element. To ensure portability and predictability of results, metafiles conforming to this specification may contain only those ESCAPE elements that are defined in 3.2.6.

3.2.1.8 **External elements**. The "action required" flag of the MESSAGE element shall be restricted to the value "no action required."

(Editor's note: Should this restriction be eliminated?)

3.2.2 Additional attribute values

3.2.2.1 **Line types**. The additional line types specified in table IV shall apply.

TABLE IV. Additional line types

LINE TYPE	CGM parameter value
single arrow	6
single dot	7
double arrow	8
stitch line	9
chain line	10
center line	11
hidden line	12
phantom line	13
break line, style 1	14
break line, style 2	15

The parameter values are those values which have been assigned by the ISO Registration Authority for Graphical Items, and are in accordance with the ISO Register of Graphical Items.

3.2.3 **FIPS PUB 128 defaults**. The defaults of all elements in this Application Profile shall be as specified in clause 6 of

Part 1 of FIPS PUB 128. Conforming basic metafiles are permitted to contain one or more METAFILE DEFAULTS REPLACEMENT elements to redefine any of these values.

3.2.4 Specification of semantic ambiguities. FIPS PUB 128 leaves the semantics of a number of graphical details unspecified or "implementation dependent." The following specifications shall apply for conforming basic generators and interpreters of this specification:

3.2.5 View surface clearing. The view surface shall be cleared upon interpretation of the BEGIN PICTURE BODY element.

3.2.6 Clipping. When the CLIP INDICATOR is "off", clipping shall be done to the intersection of the device viewport and the device view surface limits. When clipping is "on", clipping shall be done to the intersection of the clip rectangle, the VDC EXTENT, the device viewport and the device view surface limits.

3.2.7 Edge centering. Drawn edges of filled-area elements shall be centered on the ideal mathematically-defined edge of the area.

3.2.8 Font specifications. The fonts in table VI are public domain fonts, available as part of NBS SP 424. All of these fonts shall be considered basic capabilities of a basic metafile conforming to this specification. Any of these fonts may appear in the FONT LIST element in a basic metafile that conforms to this specification. Font name shall be the concatenation of the string "HERSHEY:", to designate one of the Hershey fonts, and a "name string" to designate the particular typeface.

TABLE VI. Basic font names

1.	HERSHEY:CARTOGRAPHIC_ROMAN
2.	HERSHEY:CARTOGRAPHIC_GREEK
3.	HERSHEY:SIMPLEX_ROMAN
4.	HERSHEY:SIMPLEX_GREEK
5.	HERSHEY:SIMPLEX_SCRIPT
6.	HERSHEY:COMPLEX_ROMAN
7.	HERSHEY:COMPLEX_GREEK
8.	HERSHEY:COMPLEX_SCRIPT
9.	HERSHEY:COMPLEX_ITALIC
10.	HERSHEY:COMPLEX_CYRILLIC
11.	HERSHEY:DUPLEX_ROMAN
12.	HERSHEY:TRIPLEX_ROMAN
13.	HERSHEY:TRIPLEX_ITALIC
14.	HERSHEY:GOTHIC_GERMAN
15.	HERSHEY:GOTHIC_ENGLISH
16.	HERSHEY:GOTHIC_ITALIAN
	HERSHEY:SYMBOL_SET_1
	HERSHEY:SYMBOL_SET_2
	HERSHEY:SYMBOL_MATH

(Editors note: before commencement of review, it should be determined whether the Hershey symbols should be included and given code assignments -- sticky issue concerning font versus glyph collection here.)

The fonts in table VIA are proprietary and trademarked. All of these fonts shall be considered basic capabilities of a basic metafile conforming to this specification. Any of these fonts may appear in the FONT LIST element in a basic metafile that conforms to conforming to this specification. Any of these fonts may appear in the FONT LIST element in a basic metafile that conforms to this specification. This specification in no way requires that implementors of this specification must license the named fonts from their trademark holders. Metric equivalents of the named fonts are widely available. Substitution by generators and interpreters of fonts which are "metrically equivalent", as explained in section X.Y.Z, shall constitute compliance with this specification.

TABLE VIA. <u>More basic font names</u>

1.	TIMES_ROMAN
2.	TIMES_ITALIC
3.	TIMES_BOLD
4.	TIMES_BOLD_ITALIC
5.	HELVETICA
6.	HELVETICA_OBLIQUE
7.	HELVETICA_BOLD
8.	HELVETICA_BOLD_OBLIQUE
9.	COURIER
10.	COURIER_BOLD
11.	COURIER_ITALIC
12.	COURIER_BOLD_ITALIC
13.	SYMBOL

(Editors note: before commencement of review, need to cite proper references for these trademark names.)

3.2.9 <u>Escape elements</u>. Support of the following ESCAPE elements shall be required in conforming basic interpreters.

None.

3.2.10 <u>Implementation dependencies</u>. This section specifies implementation dependencies and environmental constraints for CGM APs conforming to this specification.

3.2.10.1 <u>General guidelines for FIPS PUB 128 elements</u>. Unless otherwise noted in this specification, the guidelines of FIPS PUB 128 Annex D shall apply to conforming basic generators and interpreters as defined in 3.1.

<u>Name</u>: METAFILE DEFAULTS REPLACEMENT

<u>Description</u>: The METAFILE DEFAULTS REPLACEMENT element shall not be partitioned. Note that FIPS PUB 128 permits multiple occurrences of this element, so that partitioning is not required. Partitioning shall be permitted for all other elements.

<u>Name</u>: COLOUR TABLE

<u>Description</u>: The COLOUR TABLE element has an unspecified effect when it appears in a picture

subsequent to any graphical primitives. If a COLOUR TABLE element defining the representation of a given color index appears in a picture, it shall appear before reference to that index by an attribute element or use of that index by a graphical primitive element (included in the latter shall be implicit use of default color index attribute values by the first occurrence of an associated primitive). Once a given color representation is defined and used, it shall not be redefined. [**Note:** These restrictions insure that interpreting systems without dynamic color update capabilities shall be able to render the intended picture accurately.]

Name: PATTERN TABLE

Description: The PATTERN TABLE element has an unspecified effect when it appears in a picture subsequent to any graphical primitives filled with the affected pattern index. If a PATTERN TABLE element defining the representation of a given pattern index appears in a picture: a) it shall appear before explicit reference to that index by any PATTERN INDEX element; or b) in the case of the default PATTERN INDEX, it shall appear before any implicit reference caused by the first occurrence of an associated filled primitive. Once a given pattern representation is defined and used, it shall not be redefined. [**Note:** These restrictions insure that interpreting systems without dynamic pattern update capabilities shall be able to render the intended picture accurately.]

3.2.11 Implementation requirements for conforming basic generators and interpreters. The specifications in this section shall augment those of FIPS PUB 128, Part 1, annex D.5, and Part 3, clause 8.

3.2.11.1 Additional generator specifications. A conforming basic interpreter shall generate pictures which accurately and correctly represent the metafile being interpreted.

3.2.11.2 Additional interpreter specifications. A conforming basic interpreter shall generate pictures which accurately and correctly represent the metafile being interpreted.

In the absence of any COLOUR TABLE elements in the metafile, conforming basic interpreters shall initialize their color tables as follows: index 0 shall be set to white; index 1 shall be set to black; and indexes 2-254 shall be set by cyclic repetition of the 8 entries specified in table VII.

TABLE VII. Default COLOUR TABLE

Index	Values	Meaning
2	(1.0,0,0)	Red
3	(0,1.0,0)	Green
4	(0,0,1.0)	Blue
5	(1.0,1.0,0)	Yellow
6	(1.0,0,1.0)	Magenta
7	(0,1.0,1.0)	Cyan
8	(0,0,0)	Black
9	(1.0,1.0,1.0)	White

Note: The values '1.0' in the preceding table denote full intensity for the appropriate component.

3.2.11.3 Minimum data structure support. The following named elements shall have basic values as defined below:

Name: Maximum Color Array Dimension

Description: The basic value for the number of color values that can appear in a color array or color list parameter shall be: 1048576 for CELL ARRAY (one 1024x1024 image); 256 for each PATTERN TABLE (a 16x16 pattern); and 2048 for the complete pattern table itself (eight 16x16 patterns); 256 for each COLOUR TABLE element (entries 0-255), and 256 for the complete color table itself. CELL ARRAY and PATTERN TABLE have color array parameters and COLOUR TABLE has a color list parameter.

Name: Maximum Point Array Length

Description: The basic value for the number of points and VDC that can appear in parameters for metafile elements shall be 1024.

Name: Maximum String Length

Description: The basic value for the length of an individual string of characters shall be: 254 for all string parameters of graphical text strings; 1024 for all others (e.g., FONT LIST) except data records; 32767 for data records.

Name: Begin Segment

Description: A maximum of 256 segments, both global segments and local segments included in the count, may be defined at any time.

Name: Bundle Table

Description: Bundle representations are not settable under this specification. To insure predictable results, interpreters and generators conforming to the CGM Application Profile of this specification shall use the default values from table VIII.

TABLE VIII. Default bundle tables

	Bundle Index				
Bundle Type	1	2	3	4	5
Line Bundle					
LINE TYPE	solid	dash	dot	dash-dot	dash-dot-dot
LINE WIDTH	1	1	1	1	1
LINE COLOUR	1	1	1	1	1
Marker Bundle					
MARKER TYPE	dot	plus	asterisk	circle	cross
MARKER SIZE	1	1	1	1	1
MARKER COLOUR	1	1	1	1	1
Text Bundle					
FONT INDEX	1	1			
TEXT PRECISION	stroke	stroke			
CHARACTER EXPANSION FACTOR	1	0.7			
CHARACTER SPACING	0	0			
TEXT COLOUR	1	1			
Fill Bundle					
INTERIOR STYLE	hatch	hatch	hatch	hatch	hatch
FILL COLOUR	1	1	1	1	1
HATCH INDEX	1	2	3	4	5
PATTERN INDEX	1	1	1	1	1
Edge Bundle					
EDGE TYPE	solid	dash	dot	dash-dot	dash-dot-dot
EDGE WIDTH	1	1	1	1	1
EDGE COLOUR	1	1	1	1	1

4. QUALITY ASSURANCE PROVISIONS

4.1 **Responsibility for inspection**. Unless otherwise specified in the contract or purchase order, the contractor is responsible for the performance of all inspection requirements (examinations and tests) as specified herein. Except as otherwise specified in the contract or purchase order, the contractor may use his own or any other facilities suitable for the performance of the inspection requirements herein, unless disapproved by the Government. The Government reserves the right to perform any of the inspections set forth in the specification where such inspections are deemed necessary to ensure that supplies and services conform to prescribed requirements.

4.2 **Responsibility for compliance**. All items shall meet all requirements of section 3. The inspection set forth in this specification shall become a part of the contractor's overall inspection system or quality program. The absence of any inspection requirements in the specification shall not relieve the contractor of the responsibility of ensuring that all products or supplies submitted to the Government for acceptance comply with all requirements of the contract. Sampling in quality conformance does not authorize submission of known defective material, either indicated or actual, nor does it commit the Government to acceptance of defective material.

4.3 **Inspection procedures**. All entities, attributes and parameter values shall be analyzed for conformance to FIPS PUB 128 and to section 3 of this specification for a conforming basic metafile. This shall be accomplished with an appropriate software utility, or conformance test suite. All conforming basic metafiles contained in a particular CGM application profile shall be displayed and checked visually for conformance to the requirements of FIPS PUB 128 and of section 3 in its entirety.

(Editors note: the preceding should be improved a little to discuss more explicitly what is being looked for in the three conformance categories: metafiles, generators, interpreters.)

4.3.1 **Font rendering**. This specification shall consider any rendering of a requested font conforming if the rendering is "metrically identical" to the font metrics of the requested font. This means that the placement and alignment of the string and the placement, size, and shape of individual characters (i.e., the drawn portions of the character cells) shall be measurably identical. This does allow a good quality filled font to be substituted for a stroked Hershey font, for example. Finally, the Hershey "fonts" are really a mixture of fonts and character sets (e.g., Greek is a character set). The requirements of this specification shall be served by providing that the necessary character sets be supported in part, and the necessary typefaces be supported in part, so that the combinations required to render the listed 16 Hershey "fonts" shall be supported in full. It is recognized that the Hershey fonts may not be of adequate quality for modern publication requirements.

4.3.2 Error processing. A conforming basic interpreter shall recover from any exception condition. If there is something which is not understood by the interpreter, then if possible that element should be skipped, appropriate error warnings generated or logged, and interpretation continue with the next element following the problem element.

5. PACKAGING

Packaging of illustration data files for delivery shall be in accordance with the requirements of MIL-STD-1840A.

6. NOTES

6.1 Intended use. This specification is designed to be incorporated into a contract to define the technical requirements to be met when it is desired to purchase illustration or picture description data (in contrast to product definition data) in digital form for use in technical illustrations and technical publications. A metafile under this specification represents illustration data in the form of a conforming basic metafile, i.e., it contains, in device-, system-, and implementation-independent form, the picture description data represented by the functions invoked through an application program interface. A CGM AP contains the allowable output primitives and attributes which may be used to compose the picture. In addition, the CGM AP of this specification specifies certain constraints on CGM generators and interpreters to remove implementation dependencies, thereby serving to ensure predictable interchange of conforming basic metafiles between clients.

6.1.1 Explanation of CGM AP. The syntactic specification in the FIPS PUB 128 is complete and unambiguous. It is, as well, redundant in the sense that there are three distinct encodings of the same functionality: binary, character, and clear text. The redundancy serves a useful purpose, as each encoding is tailored to certain computing environments and applications, and so the CGM client has the opportunity to choose a syntax that is optimized to the intended application. The binary encoding has been chosen as the only encoding which will be supported by this military specification at this time.

The semantic specification is less complete. The expected overall results of using the geometric primitive elements are well enough specified. However some of the finer details, such as the precise appearance of joints and endpoints in lines, are unspecified. This underspecification of semantics was intentional on the part of the standards committees formulating the CGM standard, since it allows a wider range of existing systems to be accommodated and makes the standard more adaptable to the various needs and philosophies of a diverse clientele.

On the other hand, the semantic ambiguity does mean that there will be no single correct interpretation of a given CGM metafile, and hence it will be difficult to unambiguously describe an intended picture using the CGM standard. This is a distinct drawback in certain application environments, such as the areas of Technical Illustration and Technical Publishing.

There are further sources of uncertainty in using CGM in an application environment. A CGM metafile is produced by a component of a graphics environment known as a "metafile generator." The content of a CGM metafile is rendered into pictures by a component known as a "metafile interpreter." FIPS PUB 128 specifically excludes standardization of the behavior of metafile generators and metafile interpreters. (Most such behavior is described as "implementation dependent.") In doing so, a certain unpredictability of results is introduced into the graphics system viewed as a whole; for example, CGM generators serving GKS (Graphical Kernel System, ANSI X3.124) clients in the product lines of two different vendors might map out-of-range attributes differently.

These two sources of ambiguity in using the CGM standard--incomplete semantics and non-specification of the behavior of generators and interpreters--do not diminish the utility of FIPS PUB 128 for technical illustration and technical publishing. It is a sound and suitable basic protocol for these areas. But they do mean that some further specification (beyond that in the published standard) is required in order for the use of the CGM standard to be effective and unambiguous.

Such a specification is precisely what an Application Profile (AP) consists of. In the case of CGM, an AP specifies:

1. complete semantics;

2. the behavior of CGM generators and CGM interpreters;

An AP specifies minimal and maximal requirements for generators and interpreters, and ties down all implementation dependencies of the CGM metafile. As the name suggests, the AP for CGM is a set of specifications appropriate to a given application environment.

6.1.2 <u>Metafile Descriptor Elements</u>. It is unclear in FIPS PUB 128 whether there should be a mandatory ordering of Metafile Descriptor elements (the grammar implies some). Addendum 1 of FIPS PUB 128 will impose such an ordering when it becomes part of the standard; METAFILE VERSION, METAFILE ELEMENT LIST, and METAFILE DESCRIPTION are the first three elements, in that order.

6.1.3 <u>Additional attribute values</u>.

6.1.4 <u>Line types</u>. The line types specified in table IV of 3.2.2.1 have been registered by ISO, the International Standards Organization, for graphics registration, and are contained in the ISO Register of Graphical Items. In table IV, the name of the line type is given, followed by the numeric value (the line type parameter) by which it is to be referenced. These values are the ISO-registered values.

6.1.5 <u>Fonts and Character Sets</u>. This Application Profile contains two character sets in the BASIC Set, ASCII and ANSI X3.134/2 ("Right Hand Part of Latin Alphabet Number 1"). The Basic Set also specifies the Hershey fonts as one of the basic font families. There is finally the requirement that the requested character set be representable in the requested font. X3.134/2 is not fully representable in the digitized databases of the original public domain versions of the Hershey fonts. For the purposes of this specification, those characters of X3.134/2 which are not contained in the original Hershey set should be rendered in a way that is consistent in style and metrics. For example, the style and metrics of a Hershey version of the character "LOWER CASE A ACCENT GRAVE" should have an obvious relationship to shose of "LOWER CASE A".

This problem does not arise in the other font families of this specification.

6.2 <u>Ordering data</u>. The contract or purchase order should specify the following:

a. Title, number, and date of this specification.

b. Whether it is meant for full color, grayscale, or monochrome interpretation. (See x.y.z)

6.3 <u>Definitions</u>.

6.3.1 Acronyms and abbreviations used in this specification. Acronyms and abbreviations used in this specification are defined as follows:

a. **ANSI** - The American National Standards Institute.

b. **AP** - Application Profile.

c. **CGM** - Computer Graphics Metafile. Synonymous with FIPS PUB 128.

d. **FIPS** - Federal Information Processing Standards.

e. **GDP** - Generalized Drawing Primitive.

f. **GKS** - Graphical Kernel System.

g. **ISO** - International Standards Organization.

h. **PUB** - Publication.

i. **SP** - Special Publication.

j. **VDC** - Virtual Device Coordinates, the coordinate system of FIPS PUB 128.

6.3.2 Application Profile. A specification that defines the use of a standard, and defines all possible data streams that conform to that profile. An AP insures interoperability of different/multiple implementations of a standard. In this context, it completely and unambiguously represents the information requirements for a particular application of digital graphics data.

6.3.3 Basic values. The subset of permissible values for parameters of a CGM element that are mandatory for conformance to this specification.

6.3.4 Computer Graphics Metafile. The specification for a mechanism for storing and transferring illustration data. Refer to FIPS PUB 128.

6.3.5 Conforming basic generator. A metafile generator that produces only conforming basic metafiles (or can be reliably commanded to function in that mode), and additionally conforms to any additional generator requirements as explained in section 3.

6.3.6 Conforming basic interpreter. An metafile interpreter that at least correctly interprets any conforming basic metafile, and conforms to any additional interpreter requirements as explained in section 3.

6.3.7 Metafile. Synonymous with CGM. A representation for the storage and transfer of graphical data and control information. This representation contains a device-independent description of one or more pictures.

6.3.8 Metafile generator. The software or hardware that creates a picture or conveys information in the CGM representation.

6.3.9 Metafile interpreter. The software or hardware that reads a CGM metafile and interprets the contents.

6.3.10 Permissible values. The range of values for a parameter of a CGM element as specified in FIPS PUB 128.

6.3.11 Vector Graphics. The presentation or storage of images as sequences of line segments.

Note: Refer to FIPS PUB 128, clause 3, for further definitions of computer graphics terms.

6.4 Subject term (keyword) listing.

Application profile
CGM
CGM metafile
Digital
FIPS PUB 128
Technical illustrations
Technical publications

Appendix A: Differences between MIL-D-28003 and MIL-D-28003A

A compendium of important differences, with rationale for the strategy and changes. To be completed in FY91 prior to commencing review of 28003A.

Appendix B: Timing and Anticipated Changes for MIL-D-28003B

Identified requirements for CGM in technical publishing that are not addressed in 28003A. When and how they will be addressed in 28003B. TO be completed in FY91 prior to commencing review of 28003A.

Custodians:
Army - CR
Navy - SH
Air Force - 24
DLA - DH

Preparing Activity
OSD-CL
(Project ILSS - 0034)

Review activities:
Army - AM
Air Force - 01,02
NSA - NS
DCA - DC
NASA - NA
Others - NBS, DOE, GPO, NCS

User activities:
OSD - IR
Army - AL, AT, AV, EA, ER, GL, ME, MI, MR, SM, TE, TM
Navy - AS, EC, OS, SA, YD
Air Force - 11, 13, 14, 17, 18, 19, 68, 79, 99

FINAL REPORT

CALS CY90 SOW TASK 4.1.2

INJECTING CALS REQUIREMENTS INTO CGM AMENDMENT PROCESS

TABLE OF CONTENTS

Inject CALS Requirements in CGM Amendments

PURPOSE

Inject CALS requirements into CGEM (Extended CGM) Amendment Process (Task 4.1.2). This task was divided into the following subtasks:

a) Monitor and attend standards meetings on the CGEM work;
b) Represent and inject CALS comments and requirments into the CGEM development process; and
c) Report to CALS DoD on the progress and results of these meetings.

The actual work had two principle aspects: (1) participating in the working meetings of the graphics and metafile experts of ANSI and ISO; and (2) the inter-meeting work of preparing and coordinating position papers, drafts of standards documents, and ballot responses.

NIST/CSL has in the past participated in the CGEM work through its CALS representative on the ANSI and ISO committees, Lofton Henderson. This participation has continued in 1990.

This report summarizes progress made at the working meetings, the current status of CGM extensions work, projected timetables for completion, and recommendations for future work. The key documents, namely the drafts for CGM Amendments 1 and 3, are included as appendices.

BACKGROUND

After six years of deliberation, circulation, balloting, and refinement the Computer Graphics Metafile (CGM) became an ANSI and a FIPS standard in 1986 (and an ISO standard in 1987). One consequence of the consensus process of drafting and refining the CGM was that the standard became a 'least common denominator" graphical metafile for the various constituents. To a large degree, this standard is the area of overlap that all participants (in its formulation) agreed upon should be in a graphical metafile. As a result it is functionally "lean."

The disadvantage of a lean CGM is that it is difficult to use the CGM efficiently in some application environments. Much useful, additional functionality (particularly to support technical illustration and publishing, and compound document exchange) was proposed for CGM during its formulation. Most of the proposals were deferred, in favor of getting a lean "first generation" CGM completed as quickly as possible. An amendment or extension process was immediately commenced to begin sorting through the proposals to enrich CGM functionality in the direction of requirements for more advanced metafile applications.

At this time three amendments (until this year they were called "addenda") to CGM are in progress:

1. **Amendment 1** -- additional output capabilities and symbol libraries;
2. **Amendment 2** -- support of 3D primitives and viewing in metafiles;
3. **Amendment 3** -- support of advanced 2D drawing capabilities for technical illustration, graphics art quality picture definition, and graphics in technical publishing.

All three are formal ISO/IEC projects. The first two were endorsed in 1986, and formal status as an ISO project for the third was pursued from 1987-1989 (principally as a consequence of the activities of predecessors of this project). In 1989 the project finally passed the SC24 NWI (New Work Item) ballot. It was finally given status by Standing Committee 24 (SC24) in 1989 and Joint Technical Committee 1 (JTC1) concurred in 1990. ANSI procedures now make adoption as ANSI standards automatic for these three projects. Thus, there is no separate ANSI technical effort (or rather, ANSI functions as a participant in the ISO project).

Amendment 3 is the most critical for CALS. Parts of Amendment 1 are important as well. Amendment 2 is of low priority (or zero priority) for CALS at present. The US has consistently been trying to get processing of Amendment 2 either stopped (its current scope seems useless, and it is consuming scarce committee resources), or redefined in a way that would make it useful in the Programmers Hierarchical Interactive Graphics System (PHIGS) and product data environments (Standard for the Exchange of Product Model Data, STEP).

In 1987, due in part to CALS Program support, the scope of Amendment 1 was defined such that some functionality important to CALS constituents (e.g., Global Segments) was included. Amendment 3 was started within ANSI primarily due to CALS efforts. In fact, progress on Amendment 3 is principally due to CALS-funded participation.

In 1988, the scope and goals of Amendment 3 were more precisely defined. A technical base document was produced, complete with encodings. Formal standing for the project within ISO was sought. The result was a study group, formed to determine the need for the Amendment, to generate a requirements statement, and to produce a New Work Item proposal (the leader of this group was again the CALS representative, Lofton Henderson). The NWI ballot passed in July 1989 and circulation and review of technical Working Drafts commenced.

Amendment 1 was expedited and completed technical processing in July 1989. Due to the document editor's tardiness the final document was not forwarded to the ISO Central Secretariat until May 1990.

In FY90 the nature of this task has shifted from generating and injecting new CALS requirements into the CGM extensions pipeline. Now the focus is on expediting the progress of the amendments and ensuring that they in fact meet approved CALS Program requirements. Both of these activities are critical. Although the ISO member nations who are working on Amendment 3 have all approved the scope and the requirements, there seems to be a tendency to either forget them and leave functionality out or to add functionality beyond the approved requirements. These tendencies increase the magnitude of the standardization effort. The latter tendency in particular threatens to stretch out the schedule significantly if it is not checked.

DISCUSSION

1. Summary of Events

The following significant activities occurred in FY90 as part of this task:

1. ISO SC24/WG3 Metafile Rapporteur Group (MRG) meeting, Olinda, Brazil, October 1989: to process international comments on 1st Working Draft of CGM Amendment 3.

2. ISO SC24/WG3 Metafile Maintenance Rapporteur Group (MMRG) meeting, Olinda, Brazil, October 1989: to produce a list of defects (or errors) in the original CGM (ISO 8632:1986), issuing interpretations of ambiguities, and preparing a set of "defect reports."

3. Document production for 2nd Working Draft of CGM Amendment 3: November 1989.

4. Meeting of ANSI X3H3.3 CGM experts, Melbourne, FL, January 1990: to decide US comments on CGM Amendment 3 2nd Working Draft:

5. Production of "US Comments" document on CGM Amendment 3 2nd Working Draft and submission to ISO: February 1990.

6. Close of JTC1 ballot on the Amendment 3 NWI, project finally and officially approved: February 1990.

7. SC24/WG3 MRG meeting, Seal Beach, CA, March 1990: to process international comments on CGM Amendment 3 2nd Working Draft.

8. Document production for Proposed Draft Amendment (PDAM) text of CGM Amendment 3: April 1990.

9. Commencement of ISO SC24 3-month PDAM ballot on CGM Amendment 3: May 1990.

10. Circulation of PDAM Amendment 3 to ANSI X3H3 for letter ballot to form US comments: April-May 1990.

11. Meeting of ANSI X3H3.3 CGM experts, Austin, TX, May 1990: to process X3H3 letter ballot results and decide US comments on CGM Amendment 3 PDAM ballot.

12. Production of "US Comments" document on CGM Amendment 3 PDAM and submission to ISO: June-August 1990.

13. Close of ISO PDAM ballot for Amendment 3: 20 August 1990.

14. SC24/WG3 MRG meeting, Berlin, FRG, September 1990: to process international comments on CGM Amendment 3 PDAM ballot and produce first-cut DAM text.

15. Second SC24/WG3 MMRG meeting, Berlin, FRG, September 1990: to consider accumulated CGM "defect" reports and requests for clarification.

2. Activities Relating to the Original CGM Standard

There have been two significant actions regarding the original CGM standard, ANS1 X3.122-1986 and ISO 8632-1987. First, the X3H3.3 CGM experts recommended, X3H3 concurred, and ANSI has approved withdrawing ANSI X3.122-1986 and replacing it with ISO 8632-1987. The technical content of the two documents is identical, but they have different layout and editorial style due to the differing requirements of ANSI and ISO. This move will eliminate the need for a separate US document editor. It will also allow automatic adoption of the ISO amendments without production of separate US documents. [NOTE: This will have some effect on MIL-D-28003, which references FIPS PUB 128 (which is identical to ANSI X3.122-1986).] The timing of completion for this largely administrative move within ANSI is still uncertain.

The second event of interest was the initial meeting in October 1989 of the ISO SC24/WG3 Metafile Maintenance Rapporteur Group (MMRG). The purpose of the MMRG is to prepare and process "defect reports" concerning the CGM standard. The standard inevitably has mistakes, contradictions, and ambiguities. The MMRG considers issues which are brought to its attention, makes decisions, and the resulting "defect reports" become part of the standard (it is possible that there is an SC24 ballot first - this procedural question is uncertain now).

The results of this MMRG meeting were mostly of an editorial nature. However, there were a few technical judgments issued which should clear up some inconsistencies in implementations. A second meeting of the MMRG was held in September 1990.

3. CGM Amendment 1

The final technical work on CGM Amendment 1 took place in July 1989. At that point editing instructions were given to the Amendment 1 document editor to produce the final Amendment 1 text. The document editor did not produce text until September 1989. It was judged that the text was not sufficiently complete to go to ISO Central Secretariat (CS) following the schedule review and editing cycle. Too many significant sections had been left as "to be completed." Therefore it was agreed that a new document would have to be produced and another review performed. This was done in late 1989 and early 1990. The final text finally went to ISO CS in May 1990.

In July it was learned that Amendment 1 was being delayed because the rapporteur of the metafile group had not yet submitted the proper reports on the final Amendment 1 meeting, a full year after the meeting. This has apparently been completed and it is hoped that Amendment 1 will finally be published by ISO in the very near future. The final text of CGM Amendment 1 is included in Appendix 1.

4. CGM Amendment 3

The following subsections review major activities on CGM Amendment 3 since September 1989.

4.1 CGM RG Meeting, Olinda, Brazil

International comments on the 1st Working Draft of CGM Amendment 3 were processed by the CGM RG. Lofton Henderson was head of the US delegation at that meeting.

Inject CALS Requirements in CGM Amendments

The 1st Working Draft was incomplete. It lacked entire technical areas that were identified in the requirements document and NWI (see the US Comments in the final report of the predecessor of this project in 1989). The major result of this meeting was a set of decisions allowing production of a 2nd Working Draft with (at least) a placeholder in every required technical area. The meeting was too short, and for some technical areas the required talent not available, to produce uniform high quality across all the technical areas.

There was unanimous agreement with the "meta-issue" raised by the US: the Amendment 3 work should be driven by requirements and the needs of the technical constituency, and should not be overly constrained by concerns of compatibility and usability with other standards such as GKS. This was an important decision.

Major areas of contention included: 3D (Amendment 2), curves, and raster. As stated earlier the US has consistently been trying to get processing of Amendment 2 either stopped (its current scope seems useless, and it is consuming scarce committee resources), or redefined in a way that would make it useful in PHIGS and product data environments (STEP). There was some sympathy for the latter at Olinda (France, Austria, Germany, US) but no one could commit the resources. Therefore the project went forth and the best the US could do was "damage control"; i.e., minimize the time spent on it, and segregate it into 4 separate parts of ISO 8639 (new parts 5-8).

In the area of text and font facilities (one of the most difficult areas in CGM interchange), the following were decided:

1. A RESTRICTED TEXT METHOD element will be added, with several options enabling precise control of the extent of text string. This solves most text interchange problems for low and medium quality text.

2. Glyph shape information will be dropped.

3. Glyph metric information will not be included in the metafile, but if needed will be expected to accompany the metafile in an external resource file.

A number of adjustments were made to the new compressed raster facilities (PEL ARRAY and TILED PEL ARRAY). The emphasis was not to invent new technology, but to incorporate and encapsulate the work of other groups. The features of ODA part 7 and its newer tiling addendum are of particular interest. There was still some dissatisfaction with the messy packaging of the raster functions in Amendment 3.

Inject CALS Requirements in CGM Amendments

While everyone agreed that the IGES curve formulations in the Amendment 3 1st Working Draft were unsuitable for CGM, it was decided to leave Curves "as is" in the 2nd Working Draft. The UK would convene a workshop in late January to examine the topic and make recommendations.

The mechanism of Geometric Pattern interior fills was defined, as was a simple mechanism for interpolated interiors. In liaison with a SC18 color expert, the color extensions of Amendment 3 were further refined. "Spot" color was thrown out altogether. Once again the metafile group is following and incorporating the work of other committees here. To a point Amendment 3 will be changed to follow SC18. Currently color models include: RGB (Red-Green-Blue), CMYK (Cyan-Magenta-Yellow-Black) and CIELAB (Commission Internacionale L'Eclairage uniform color space).

Amendment 3 was caught in a procedural bind that prevented the project from officially commencing. Although the SC24 ballot had completed in July, the JTC1 ballot was delayed. Although a formality, the project could not be given official designation until the JTC1 ballot closed in February. For this reason it was agreed at Olinda to do a 2nd Working Draft circulation, and process the resulting comments at an MRG meeting in March. This meeting would have authority to commence simultaneous PDAM Registration and PDAM ballots.

4.2 US Metafile Experts Meeting, Melbourne, FL

In late January 1990 there was a regularly scheduled X3H3 meeting in Melbourne, FL. The X3H3.3 metafile subgroup met for 4 days to consider the Amendment 3 2nd Working Draft and assemble the US comments. These comments comprised more than 60 issues which were identified and argued. The group reached consensus on a number of these, while others were left open. The US position was assembled from these issues and submitted. Part of the US position was that all of these issues had to be resolved before the US could approve PDAM status.

Once again, the major issues were Raster and Curves. A preview of the UK position indicated that the UK would propose "NURBS only" for the advanced curves elements of Amendment 3. The US considered this to be contrary to current engineering practice, and unnecessarily expensive for applications needing only a simple Bezier curve or a conic arc.

A number of complicated and inter-related conceptual issues regarding the formulation of compressed raster elements were dealt with, and a tentative set of recommedations approved. In particular, a thorny issue arose of whether the compression techniques were:

1. defined at the functional level so that each compression technique is instantiated in each encoding; or

2. only defined at the encoding level, with different sets available in each encoding.

The resolution was for alternative 1. The reasons were to be able to inter-translate encodings as well as to pass pre-compressed raster arrays into the metafile in a straightforward manner. It is a contentious issue however, and has involved much time at subsequent national and international meetings.

The technical area of External Symbols was recognized as requiring a lot of work. A recommendation that it be properly sorted out was generated along with a number of open issues to guide the sorting out process. A proposal for accessing families of typographic quality glyphs was prepared and submitted.

4.3 CGM RG Meeting, Seal Beach, CA

Major areas of contention included: Curves; Symbols; Filling Mechanisms; Raster; Transformations (Picture Mapping). In summary, good progress was made at this meeting. The major technical components of Amendment 3 began to stabilize. Amendment 3 is still on the aggressive schedule proposed and agreed at Waikoloa (July 1989) and Olinda (October 1989).

4.4 PDAM Ballot and Austin, TX Meeting

Appendix 2 contains the CGM Amendment 3 PDAM text that resulted from decisions taken at the Seal Beach meeting. This was the subject of an SC24 ballot which completed in August. An X3H3 letter ballot was conducted in order to gather input for the US position and comments on the PDAM text. The metafile experts of X3H3.3 met for four days in Austin, TX in May 1990 to process the results of the X3H3 letter ballot and produce a US position. Some 130 issues were raised on the letter ballot, and all were addressed to some degree at Austin. A handful were carried over until after the meeting.

The US PDAM position was the major result of this meeting. Briefly, the areas of major contention and major results were:

Inject CALS Requirements in CGM Amendments

1. A complete proposal was prepared. It completely rewrites the raster section, finally giving a clean packaging of the compressed, tiled raster capabilities;

2. More clarification on the workings of path text, and more pictures and examples were prepared and submitted;

3. The External Symbols mechanism was further cleaned up and adjusted;

4. A justification of the need for the Picture Mapping transformation was requested;

5. Adjustments to filling mechanisms were generated. These included:
   - o generalize interpolated fill, to account for current practice in currently available presentation products;
   - o addition of smooth-shaded triangular facets;
   - o removal of complexity added to user-defined hatch at Seal Beach.

6. Transformations: There is currently a fair amount of confusion in CGM, CGI, and Amendment 3 as to what things transform and what don't. For engineering practice, hatch and linewidth should not transform arbitrarily. A contribution was completed and submitted with the US comments. It tried to sort out a method for meeting all identified requirements cleanly.

7. Some fine tuning of the formulation of curves was proposed.

The result of the Austin meeting was that the technical content had fairly well stabilized. Following the next round of review and revision Amendment 3 should be approaching stability in formulation and presentation as well. This was the case within the US review process, but there was some uncertainty concerning its stability in the ISO reviews. Indications were that the UK also wanted to keep things simple and minimal to get the work completed. There was worry that Germany was taking a somewhat "academic" and unpragmatic attitude, and may have wanted to include facilities that would take a long time to define and work out. France would be participating for the first time at the Berlin meeting in September, and there was very real concern that they may want to slow the work significantly. Japan was also to participate for the first time, and there was no idea what to expect from them.

4.5 CGM RG Meeting, Berlin, FRG

This meeting was conducted from 17-20 September. In summary, the meeting went well and the major technical components of Amendment 3 have stabilized. There were a couple of major adjustments made, particularly regarding questions of transformations and specification modes. Hopefully these will be seen both in the US and other nations as necessary and appropriate.

The conclusion of the meeting was that Amendment 3 should advance to DAM ballot. This is a 6-month ISO ballot, which should be able to commence sometime in December 1990. Assuming this schedule, and assuming that the Amendment 3 project can be closed with a single DAM ballot, then the final processing of Amendment 3 should be accomplished in the summer of 1991. The aggressive schedule proposed and agreed at Waikoloa (July 1989) and Olinda (October 1989) will have been maintained. There will be two ANSI meetings, a letter ballot, and a DAM ballot position preparation during the 6-month period of the DAM ballot.

There was a week of editing activity at Berlin following the formal close of the metafile group meeting (the NIST Representative is now the document editor). This produced substantially complete DAM text. It is being reviewed by the meeting participants and simultaneously worked on by the document editor. The DAM text should be forwarded to the SC24 secretariat for initiation of the DAM ballot in early November.

As mentioned elsewhere in this report, there has been a tendency for some of the other nations to complicate and aggrandize the project beyond the necessity of meeting the requirements. Preventing such growth of the project and the delays which would be associated has been one of the major tasks of the US delegation (and the NIST representative) at meetings in 1990. For the first time at the Berlin meeting it was perceived that the other participants largely shared the motive to stick closely to the agreed requirements and finish the work in as concise and timely a manner as possible. If such an attitude is maintained, the aggressive schedule for completion of the technical work in summer of 1991 will be maintained.

CALS USE AND IMPACT

Amendment 1 required little activity in 1990. Little more was required than to monitor its progress through its final procedural steps. Unfortunately, the Amendment 1 document editor and former rapporteur were slow in producing the document and completing final reports. Amendment 1 could (in the best of circumstances) have been published in late 1989 or early 1990. However, it should soon be published now that the final procedural requirements have been completed.

Progress was excellent on Amendment 3. The document moved from First Working Draft, to Second Working Draft, to PDAM text, to DAM text in the calendar year which elapsed since the close of the Olinda meeting. Such rapid progression is unusual for ISO standards work.

Virtually all of the technical goals which NIST/CSL has worked for in support of CALS Program requirements have been achieved in Amendment 1 and are hopefully less than one year from achievement in Amendment 3.

The functionality that is reaching stability in this process has been the input for revisions to MIL-D-28003. Some will appear in Revision A, while more will appear in Revision B in a couple of years or so.

SUMMARY AND CONCLUSIONS

Previous work in this area has focussed on defining the CALS requirements for CGEM, getting them endorsed by ANSI, and introducing them into the ISO CGM amendment processing. The requirements definition, domestic and international endorsement of the requirements, and international acceptance of a project to implement these requirements have been accomplished.

In 1990 the first completed result of the CGEM project, CGM Amendment 1, has been sent to the ISO Central Secretariat for final processing to become an International Standard. Work on getting functionality into CGM Amendment 1 that meets some of the CALS needs is finished. The only activity related to Amendment 1 has been communication with both the document editor and the former rapporteur of the ISO SC24/WG3 Metafile Rapporteur group. This ensured that the final text was produced and forwarded and that the final procedural steps were completed in order for the Amendment to be published.

The major uncompleted work included: the formality of completing the ISO endorsement for further extensions the Amendment 3 project; and the more significant task of advancing the technical work already underway on the project.

Specific goals for FY90 were:

1. Monitor and expedite publication of CGM Amendment 1.

Inject CALS Requirements in CGM Amendments

2. Advance Amendment 3 through the formal technical processing steps both within the US committees and ISO committees.

At the end of FY89 a first Working Draft for Amendment 3 had been produced. At the end of FY90 the DAM (Draft Amendment) text of Amendment 3 is nearly complete and what hopefully will be the last ISO balloting and review period, is about to commence. Two complete cycles of national and international review have occurred in the last calendar year. According to the original schedule in the New Work Item, final text was anticipated in April 1991. It is now estimated that it will be August 1991.

The progress of the work has been excellent during the contract period. The content and quality of the work (from a CALS perspective) has been maintained as well. The major effort that CALS has made (through its NIST representative) has been getting the CALS requirements endorsed, and ensuring that the technical work actually implemented those requirements. The goals for FY90 were achieved.

RECOMMENDATIONS

The excellent progress on CGM Amendment 3, the "CALS Amendment," has had a cost. Unfavorable economic conditions in the US have forced several senior participants in X3H3.3 to terminate their participation. Among these are two participants who contributed significantly to document production and meeting organization. The 1989 document editor is officially gone as well. Virtually all necessary roles in the US metafile committee, including X3H3.3/CGM Task Group Leader, US Head of Delegation to ISO metafile group meetings, and Document Editor for Amendment 3 are now being fulfilled by the NIST/CSL CALS representative, Lofton Henderson.

It is unfortunate that the work is not more evenly spread among X3H3.3/CGM volunteers. Nevertheless it is the reality of the situation and it will not likely improve in 1991. If Amendment 3 is going to complete on its aggressive schedule, it will require the continued participation of this representative at the current level of effort through the summer of 1991.

Recommendation: CALS DoD should continue to fund NIST/NCSL through 1991 to expedite completion of the formal processing of CCM Amendment 3.

GLOSSARY

AFNOR	The French organization for standards work.
ANS	American National Standard, the final stage in the ANSl pipeline, nothing remains but possibly the printing.
ANSI	American National Standards Institute.
ASC X3H3	Accredited Standards Committee X3H3, the ANSI accredited committee responsible for computer graphics standards in the US.
BSI	British Standards Institute, the British organization for standards work.
CGEM	Computer Graphics Extended Metafile, a set of addenda and extensions to CGM, being processed by ISO.
CGI	Computer Graphics Interface, another ANSI/ISO standards project, currently at the DIS stage. CGI is an interface standard which exists about at the level of the CGM in the graphics pipeline (device level). CGI is an interactive (input) and highly extended and enriched interface specification, whereas CGM has output-only functionality (for picture definition) and is a picture description protocol (a graphical database). CGI embeds CGM output functionality as a subset.
CGM	Computer Graphics Metafile, ANSI standard X3.122-1986 and ISO standard ISO 8632/1-4 1987.
CS	Central Secretariat of ISO.
DAM	Draft Amendment, the same as DIS, but for an amendment as opposed to a standalone project.
DIN	The German organization for standards work.

DIS	Draft International Standard, the project stage in the ISO pipeline after DP. The technical content of the project is supposedly highly stable and it is expected that IS text can be produced subsequent to processing the DIS ballot results.
DP	Draft Proposal, the second stage in the ISO processing pipeline. After national bodies have commented on the WD, it is altered and refined and then registered as a DP. Another round of ballot and comment takes place on the DP.
GKS	Graphical Kernel System, an application programmer interface to computer graphics, now an ANSI and ISO standard.
GKSM	A metafile for use with GKS. One was proposed in non-standard Annex E of GKS. Work on it was deferred in favor of CGM, and now of extended CGM (CGEM).
Glyph	An identified abstract graphical symbol independent of any actual image. It replaces the term "character" when discussing codes and character sets. A glyph collection, then, is a precise substitute for the loosely defined term "font."
FIPS	Federal Information Processing Standards.
IS	International Standard, the final stage in the ISO pipeline, nothing remains but possibly the printing.
ISO/IEC JTC1/SC24	International Organization for Standardization/ International Electrotechnical Commission, Joint Technical Committee 1/ Standing Committee 24, the international counterpart to X3H3.
ISO TC97/SC21/WG2	The predecessor to SC24 (prior to December 1987).

MMRG	Metafile Maintenance Rapporteur Group, the subgroup of WG3 resonsible for CGM maintenance.
MRG	Metafile Rapporteur Group, the subgroup of WG3 resonsible for CGM review and CGM extensions.
NWI	New Work Item.
PDAD	Proposed Draft Amendment, the same as DP, but for an addendum as opposed to a standalone project.
PDAM	Proposed Draft Amendment, the same as DP, but for an amendment as opposed to a standalone project.
PHIGS	Programmers Hierarchical Interactive Graphics System, an application programmer interface to computer graphics, with 3D, structure hierarchy, etc., meant to be highly dynamic. It is an ANSI and ISO standard.
RG	Rapporteur Group.
STEP	Standard for the Exchange of Product Model Data.
WD	Working Draft, the first complete draft of a proposed ISO standard, the starting document for subsequent work and review
WG3	The working group of SC94 responsible for standards work in metafiles and device-level interfaces, i.e., CGM and CGI.
X3H3.3	The subcommittee of X3H3 that is responsible for CGM and CGI.

APPENDIX 1

FINAL TEXT OF CGM AMENDMENT 1

ISO/IEC 8632-1 : 1987/Am.1 : 1990

Information processing systems - Computer graphics - Metafile for the storage and transfer of picture descriptive information -

Part 1: Functional specification

Amendment 1

Page 1

Add the following at the end of 0.1:

This picture description includes the capability for describing static pictures. Static pictures are those where elements which may lead to dynamic effects (for example those leading to regeneration) are prohibited within the picture body.

Page 1

Sub-clause 0.3: Add the following at the end of item c):

It should also not preclude further extensions to support future standards.

Page 1

Sub-clause 0.3:Add the following at the end of item d):

It should include the capability to support ISO 7942 (GKS) static picture-capture.

Page 3

Add the following at the end of 0.8:

There is a very close relationship between many of the elements in ISO 8632 and a subset of the functions in the CGI (Computer Graphics Interface - ISO/IEC 9636 (currently a Draft International Standard)).

Page 4

Clause 1: Add the following at the end of the first paragraph:

This picture description includes the capability for describing static images.

Page 5

Clause 2: Add the following to the list of references:

ISO/IEC 9636 *Information processing systems - Computer Graphics - Interfacing techniques for dialogues with graphical devices (CGI).* Parts 1-6 (currently a Draft International Standard).

Page 6

Clause 3: Add the following to the list of definitions and abbreviations:

3.1.49 anisotropic mapping: A mapping in which the scale factors applied along each axis are not equal. This is often used in reference to the mapping from VDC to distance units on the physical display surface. With anisotropic mapping, the angle between any pair of non-parallel line segments can change; circles cease to be circles and become post-transformed ellipses. See "isotropic mapping".

3.1.50 boundary: The mathematical locus that defines, in abstract VDC space, the limits of a region to be filled (for fill primitives and closed figures). The visual appearance of interior style 'hollow' consists of a depiction of the boundary obtained after clipping has been taken into account.

3.1.51 character set: The set of displayable symbols mapped to individual characters in a TEXT, APPEND TEXT, or RESTRICTED TEXT string. This corresponds to the "G-set" defined in ISO 2022. A character set is independent of the font or typeface; examples of character sets are: ASCII (X3.4), German and Katakana.

3.1.52 clipping mode: A generic term referring to one of Line Clipping, Marker Clipping or Edge Clipping Modes. An object clipping may be either 'locus', 'shape' or 'locus then shape'.

3.1.53 closed figure: A compound primitive that behaves as a fill primitive of more general shape. It is formed by bracketing a sequence of line or fill primitives, edge attributes, and certain control elements, with the elements BEGIN FIGURE and END FIGURE.

3.1.54 compound primitive: A compound primitive is specified by a sequence of CGM elements, as opposed to primitives represented by a single element. Compound text and closed figures are examples of compound primitives in the CGM.

3.1.55 compound text: A compound text primitive is formed through the use of APPEND TEXT. There may be attribute changes between portions of the resulting complete text string.

3.1.56 device coordinates: The coordinates native to a device; device-dependent coordinates; physical device coordinates.

3.1.57 device viewport: A rectangular subset of the physical display surface into which VDC EXTENT is mapped. See "effective viewport".

3.1.58 edge: The rendering of the perimiter of a filled region, controlled by edge attributes. Edges are clipped after being applied to the boundary, as distinct from the rendition of the boundary obtained from interior style 'hollow'. See "boundary".

3.1.59 effective viewport: The actual viewport resulting from forced isotropic mapping from the VDC extent to the viewport.

3.1.60 foreground colour: The colour used in the rendering process in which primitives are rendered on the display surface, as opposed to the BACKGROUND COLOUR or AUXILIARY COLOUR. The foreground colour is set separately for each class of primitive.

3.1.61 global segment: A segment that is defined in the Metafile Descriptor (see "segment"). It may be referenced from within any picture.

3.1.62 graphic object: A graphic object is a graphic primitive, including a compound primitive, together with the associated attributes.

3.1.63 isotropic mapping: A mapping which is invariant with respect to direction; equal scaling in all orthogonal representational dimensions. It is often used to describe the mapping from VDC to distance units on the physical display surface. With isotropic mapping, the angle between any pair of non-parallel line segments remains unchanged; for example, circles remain circles. See "anisotropic mapping".

3.1.64 local segment: A segment whose definition is local to the picture in which it appears.

3.1.65 object clipping: Object clipping is applied to a graphic object. For example, clipping is applied to a line after it has had the width attribute associated with it.

3.1.66 region: In the context of closed figures or the POLYGON SET element, an area that is explicitly or implicitly closed, that is a subset of the full area being filled. Regions can be nested, disjoint or overlapping. The boundaries of all regions are considered together when applying the interior test for filling a closed figure or POLYGON SET.

3.1.67 segment: A collection of primitives, primitive attributes and some additional attributes associated with the segment as a whole. See "segment attribute".

3.1.68 segment attribute: An attribute associated with a segment as a whole rather than attributes of individual primitives.

3.1.69 size specification mode: A generic term for Line Width Specification Mode, Edge Width Specification Mode, or Marker Size Specification Mode. A size specification mode may be 'absolute' or 'scaled', the latter being referenced to a nominal size in device coordinate space.

3.1.70 skewed: Used to describe stroke precision text when the CHARACTER ORIENTATION vectors are non-perpendicular; CELL ARRAYs when the three defining points form a parallelogram which is not a rectangle; or a segment transformation that causes rectangles to become non-rectangular parallelograms.

Page 7

Sub-clause 3.1.26: Definition of graphical elements

Insert "primitive" between "graphical" and "element".

Page 9

Sub-clause 4.1: Add the following at the end of the list of classes of elements:

- Segment Elements, which enable the grouping and manipulation of elements.

Page 9

Sub-clause 4.1: Add the following after the third paragraph:

Graphical output primitives and attributes may be grouped in segments. Segment attribute elements control the appearance of segments.

Page 10

Sub-clause 4.2: Add the following at the end:

Primitives may be grouped together to form a composite primitive known as a closed figure. The primitives to be included in the closed figure being defined are delimited by the elements BEGIN FIGURE and END FIGURE.

Groups of elements, called segments, are delimited by BEGIN SEGMENT and END SEGMENT. Each segment is uniquely identified by a segment identifier. Segments may be defined in the Metafile Descriptor or within picture bodies.

Page 10

Sub-clause 4.3: Add the following to the list after the first paragraph:

NAME PRECISION
MAXIMUM VDC EXTENT
SEGMENT PRIORITY EXTENT

NOTE - Other elements, as defined in this part of ISO/IEC 8632, may appear within the Metafile Descriptor within the definition of a global segment.

Page 10

Add the following paragraph at the end of 4.3:

METAFILE VERSION and METAFILE ELEMENT LIST shall occur only once in the Metafile Descriptor for version 2 metafiles. It is recommended that they shall only appear once in version 1 metafiles.

NOTE - It is recommended that the following elements: METAFILE VERSION, METAFILE ELEMENT LIST and (possibly multiple occurrences of) METAFILE DESCRIPTION appear first in the Metafile Descriptor and in the order listed.

Page 10

Sub-clause 4.3.2 : Change the start of the third sentence from "Two shorthand names....." to:

Several shorthand names.......

Page 11

Add the following after 4.3.2.2:

4.3.2.3 Version 2 set

The Version-2 set may be used to indicate all the elements in the drawing-plus-control set and all the additional elements defined in this part of ISO/IEC 8632.

4.3.2.4 Extended primitives set

The extended-primitives set may be used to indicate those primitives which are not defined in ISO 7942 (GKS). These elements are:

DISJOINT POLYLINE
RESTRICTED TEXT
APPEND TEXT
POLYGON SET
RECTANGLE
CIRCLE
CIRCULAR ARC 3 POINT
CIRCULAR ARC 3 POINT CLOSE
CIRCULAR ARC CENTRE
CIRCULAR ARC CENTRE CLOSE
CIRCULAR ARC CENTRE REVERSED
ELLIPSE
ELLIPTICAL ARC
ELLIPTICAL ARC CLOSE
CONNECTING EDGE

4.3.2.5 Version 2 GKSM set

The Version-2-GKSM set includes elements for ISO 7942 (GKS) picture capture. The elements included in the Version-2-GKSM set are:

BEGIN METAFILE
BEGIN PICTURE
BEGIN PICTURE BODY
END PICTURE
BEGIN SEGMENT
END SEGMENT
END METAFILE
METAFILE VERSION
METAFILE DESCRIPTION
VDC TYPE
INTEGER PRECISION
REAL PRECISION
INDEX PRECISION
COLOUR PRECISION
COLOUR INDEX PRECISION
NAME PRECISION
MAXIMUM COLOUR INDEX
COLOUR VALUE EXTENT
METAFILE ELEMENT LIST
METAFILE DEFAULTS REPLACEMENT
FONT LIST
CHARACTER SET LIST
CHARACTER CODING ANNOUNCER
MAXIMUM VDC EXTENT
SEGMENT PRIORITY EXTENT
VDC EXTENT
DEVICE VIEWPORT
DEVICE VIEWPORT MAPPING
DEVICE VIEWPORT SPECIFICATION MODE
LINE REPRESENTATION

MARKER REPRESENTATION
TEXT REPRESENTATION
FILL REPRESENTATION
VDC INTEGER PRECISION
VDC REAL PRECISION
CLIP RECTANGLE
POLYLINE
POLYMARKER
TEXT
POLYGON
CELL ARRAY
GDP
LINE BUNDLE INDEX
LINE TYPE
LINE WIDTH
LINE COLOUR
MARKER BUNDLE INDEX
MARKER TYPE
MARKER SIZE
MARKER COLOUR
TEXT BUNDLE INDEX
TEXT FONT INDEX
TEXT PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR
CHARACTER HEIGHT
CHARACTER ORIENTATION
TEXT PATH
TEXT ALIGNMENT
CHARACTER SET INDEX
ALTERNATE CHARACTER SET INDEX
FILL BUNDLE INDEX
INTERIOR STYLE
FILL COLOUR
HATCH INDEX
PATTERN INDEX
FILL REFERENCE POINT
PATTERN TABLE
PATTERN SIZE
COLOUR TABLE
ASPECT SOURCE FLAGS
PICK IDENTIFIER
ESCAPE
MESSAGE
APPLICATION DATA
SEGMENT TRANSFORMATION
SEGMENT HIGHLIGHTING
SEGMENT DISPLAY PRIORITY
SEGMENT PICK PRIORITY

Page 12

Sub-clause 4.4. Add the following text at the end of the first paragraph:

Some of the picture descriptor elements may appear outside the Picture Descriptor if this is permitted by the formal grammar for the metafile version. In such a case they do not set parameter values to apply for the entire picture.

Page 12

Sub-clause 4.4.2. Change the text to the following:

COLOUR SELECTION MODE selects either indexed or direct (RGB) colour specification and is described further under colour attributes. For version 1 metafiles the selection is for the whole picture.

Page 12

Add the following paragraph at the end of 4.4.4:

MAXIMUM VDC EXTENT defines an extent which bounds the VDC extent values which may be found in the metafile. It may be, but need not be, a closest bound in the sense that it exactly equals the union of the extent rectangles in the metafile. This element may be used, for example, to map integer virtual device coordinates of the metafile to a unit square in a normalized device space.

Page 14

Add the following after 4.4.6:

4.4.7 Device viewport control

The device viewport specifies the region of the device display surface into which the VDC extent is to be mapped on interpretation. VDC-to-Device mapping is determinied by the VDC extent, device viewport, and device viewport mapping.

The position of the device viewport is specified in one of three coordinate systems selected by the DEVICE VIEWPORT SPECIFICATION MODE element:

- by fraction [0.0 to 1.0] of the available display surface, which allows reasonable placement and relative sizing of the viewport;
- in millimetres times a scale factor, which allows absolute sizing of images;
- in physical device coordinates.

The device viewport is specified in terms of two points on the device display surface at diagonally opposite corners of the rectangle. Mirroring or 180^{0} rotation of the image may be achieved by specifying the corners in some way other than the first as below and to the left of the second.

The DEVICE VIEWPORT MAPPING element may be used to force isotropic mapping even if the specified VDC extent and device viewport would not otherwise have led to one. In such a case, the VDC extent is mapped on to a subset of the specified device viewport on interpretation. This subset is defined by shrinking either the vertical or horizontal dimension of the specified viewport as needed to reach the required aspect ratio. This smaller "effective viewport" is then used to define the coordinate mapping from VDC to the device's coordinates. The placement of the effective viewport rectangle within the original one can be specified. This placement can be one of 'left', 'right' or 'centred' when the shrinking is horizontal, and 'top', 'bottom' or 'centred' when it is vertical. These meanings are relative to the display surface of the device.

The VDC-to-Device mapping maps the first point specifying the VDC extent on to the corner of the effective viewport corresponding to the first point specifying the device viewport, and similarly for the second point. The mapping is linear in each dimension, but is not necessarily isotropic (for example, a circle in VDC may not appear as a circle to the viewer).

Both the way VDC space is oriented relative to the display surface and the way the effective viewport is placed on the physical device may lead to mirroring and 180^{0} rotation.

The behaviour of primitives and attributes with significance in VDC space under transformations is further described in 4.6.

If both device viewport and scaling mode appear in the same metafile then the last specified is used. If neither appear then the default values for device viewport take precedence.

4.4.8 Representations

The elements LINE REPRESENTATION, MARKER REPRESENTATION, TEXT REPRESENTATION, FILL REPRESENTATION and EDGE REPRESENTATION are used to set all of the attribute values in a bundle table entry at the same time. The attributes that may be bundled are described in 4.7.

Page 14

Add the following at the end of 4.5:

Some of the control elements may appear in the Picture Descriptor if this is permitted by the formal grammar for the metafile version.

Page 15

Add the following text at the end of 4.5.2:

There are three different clipping modes for lines, markers and edges. The required clipping mode is recorded in the metafile with the elements: LINE CLIPPING MODE, MARKER CLIPPING MODE , and EDGE CLIPPING MODE. When the CLIP INDICATOR associated with a graphical primitive is 'on', only those parts of a graphical primitive that are considered inside the effective clipping region are rendered on interpretation. The object clipping modes allow precise specification as to how clipping is applied to primitives on interpretation.

Clipping may be either 'locus', 'shape' or 'locus then shape'. Conceptually, a locus is a mathematical object like a point or line segment, while a shape is an area in 2-dimensional space. Loci are 0-, 1- or 2-dimensional subsets of real-valued 2-space. For markers and text they are points. For lines they are the individual line segments or portions of arcs. The locus of an area is the shape and the boundary. Shapes reflect the realization of geometric attributes and are generally 2-dimensional subsets of real-valued 2-space.

'Locus' clipping is applied for each portion of a graphic object based on its mathematical location and is independent of the area it will occupy after rendering. For example, no portion of a line segment is rendered if the ideal mathematical line lies outside the effective clipping region (even if its line width would carry some portion of the rendering of it into the clipping rectangle); no portion of a marker is rendered if its location lies outside the clipping rectangle.

If 'locus' clipping is used, the rendering is applied to the locus of the graphic object after clipping. The resulting rendered shape areas may therefore extend outside the effective clipping region.

'Shape' clipping is applied after the abstract rendering of shape in device coordinate space. The 2-dimensional point set associated with the graphic object is intersected with the effective clipping region, which has been transformed to device coordinate space.

'Locus then shape' clipping allows the specification that both 'locus' and 'shape' clipping be applied to graphic objects as described above. In this case however, the rendered shape will not extend outside the effective clipping region. A thick line whose locus is outside the clip rectangle will not have any portion visible even if its line width would carry some portion of the rendering inside the clip rectangle.

Figure 1a shows some examples of the effect of the clipping modes.

When a width or size specification mode is 'scaled', the rendering of shape proceeds in device coordinate space after application of the VDC-to-Device mapping.

When a width or size specification mode is 'absolute', the rendering of shape proceeds, conceptually, in VDC space before application of the copy transformation, before application of the segment transformation and before the VDC-to-Device mapping.

Fill and text primitives do not have associated object clipping modes (though the edge of a fill primitive and the boundary edges of a closed figure do). Clipping for fill primitives is always consistent with 'shape' clipping (see 4.6.4.5). For text primitives, the type of clipping is determined by the associated text precision:

For 'string' precision text, clipping proceeds, on a per string basis, in a manner consistent with 'locus' clipping.

For 'character' precision text, clipping proceeds, on a per character basis, in a manner consistent with 'locus' clipping.

For 'stroke' precision text, the clipping always proceeds in a manner consistent with 'shape' clipping.

NOTE - 'shape' clipping for all text precisions is always allowed by this part of ISO/IEC 8632.

Clip rectangles applied to graphical primitive elements within segments may be subject to transformations in VDC space. Intersection of clip rectangles (untransformed or transformed) may result in polygonal clipping boundaries (see 4.12.5).

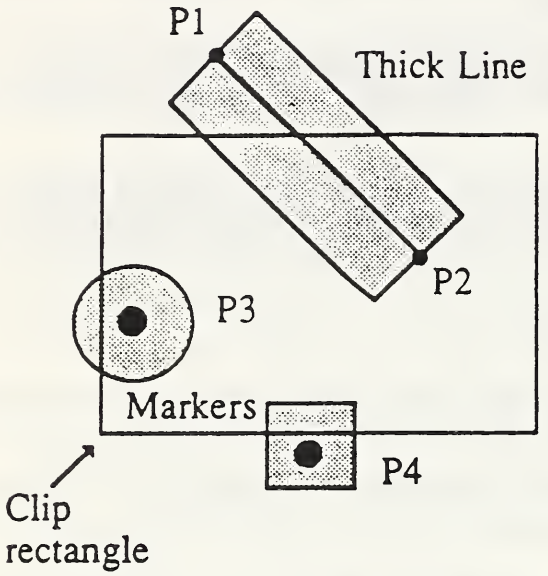
Page 15

Add the following after 4.5.2

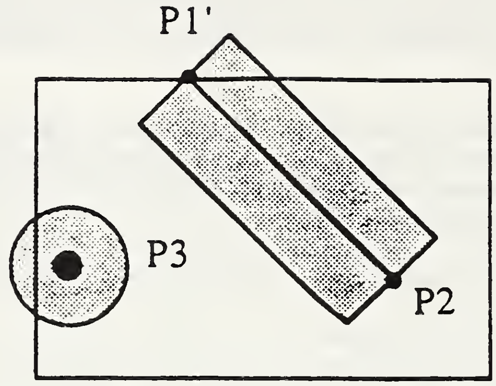
4.5.3 Save and restore primitive context

Two elements are provided to save and restore a context; that is, attributes and control elements as collections. This capability allows a list of attributes and control elements (see 5.5.11) to be stored in the metafile which can be referenced by name at a later point in the metafile. This capability can be used to save and restore attributes and control elements in conjunction with opening and closing segments.

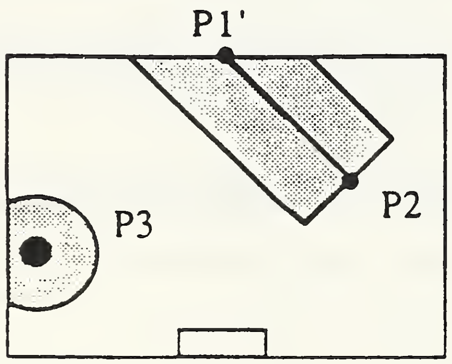
The values for attributes controlled by specification or selection modes are saved in the mode in which they were last specified along with the value of the corresponding mode. In restoring a context the current specification and selection modes are not changed.



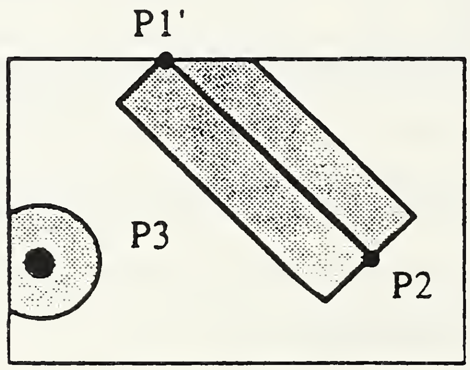
Primitives and clip rectangle stored in the CGM



Picture resulting from 'locus' clipping modes



Picture resulting from 'shape' clipping modes



Picture resulting from 'locus then shape' clipping modes

Figure 1a - Examples of the effects of object clipping modes

Page 15

Add the following to the list of graphical primitive elements and to the list of line elements in sub-clause 4.6:

CIRCULAR ARC CENTRE REVERSED
CONNECTING EDGE

Page 16

Add the following before sub-clause 4.6.1:

In addition to the graphical primitive elements listed above, this part of ISO/IEC 8632 defines elements permitting the definition of 'compound primitives' from several of the other graphical primitives. The following classes of compound primitives are defined: 'compound text' and 'closed figures'. The elements that may be used to specify compound primitives are listed in table 1a.

Table 1a - Contributing primitives to compound primitives

Compound Primitive	First Element	Primitives Included	Other Elements	Final Element
Compound Text	TEXT RESTRICTED TEXT (Note 1)	APPEND TEXT (Note2) GDP (Note 5)		APPEND TEXT (Note 3) GDP (Note 5)
Closed Figure	BEGIN FIGURE	Line Primitives Fill Primitives (Note 4) GDP (Note 5)	NEW REGION	END FIGURE

NOTES

1 The final/not final flag is 'not final'; the primitive defines the reference point of the entire compound text primitive; the text of the primitive is accumulated.

2 The final/not final flag is 'not final'.

3 The final/not final flag is 'final'; the text of the primitive is accumulated before the compound primitive is closed.

4 All primitives of the identified classes may be included.

5 Whether a GDP may contribute to compound text or closed figures, and whether or how it specifies that the compound text state or closed figure state be opened, maintained or closed, is specified with the definition of the GDP in the International Register of Graphical Items.

Graphical primitive elements and compound primitive elements may be subject to transformation in VDC space (segment and copy transformation, see 4.12.4.2 and 4.12.5). Such a transformation may change the shape of some primitives. If there is a skew, a primitive initially specified as a rectangle may become a parallelogram. If there is an anisotropic scaling, a primitive initially specified as a circle may become an ellipse. Note that the shape of markers is not affected by such transformations. Anisotropic transformation will change the angle at which non-parallel lines intersect; isotropic transformation will preserve the angle at which non-parallel lines intersect.

Page 16

Sub-clause 4.6.1.1. Add the following text to the paragraph describing CIRCULAR ARC xxx:

A reverse direction arc can also be specified; see 5.6.20.

Page 16

Add the following at the end of 4.6.1.1:

CONNECTING EDGE A line segment connecting the last point of the preceding line element to the next point is generated during the construction of a closed figure. The next point is either the first point of the next line element or the current closure point.

Page 16

Add the following at the end of 4.6.1.3:

In version 2 metafiles, line clipping is controlled by the LINE CLIPPING MODE element, which can have one of the following values: 'locus', 'shape', or 'locus then shape'. However, clipping applies only if the CLIP INDICATOR is 'on'.

For 'locus' clipping, the mathematical locus of the line is clipped at the intersection with the clip rectangle before shape rendering is applied. Hence, part of the shape of a clipped line may appear outside the clip rectangle.

For 'shape' clipping, the shape of the rendered line is clipped to the intersection with the clip rectangle; that is, nothing is drawn outside the clip rectangle. A portion of a widened line may appear inside the clip rectangle even though the mathematical locus of the line itself may be entirely outside the clip rectangle.

For 'locus then shape' clipping, the mathematical locus of the line is clipped, as with locus clipping, and then subsequently the rendered shape of the clipped locus is again clipped. Note that, since the mathematical locus of the line may have changed as a result of locus clipping, subsequent shape rendering and clipping may produce a different appearance of a line from either of the other two clipping modes.

If the line width is measured in VDC units it is subject to the VDC-to-Device mapping (4.4.7) as well as to both segment and copy transformation (4.12.4.2 and 4.12.5). Note that the entire locus of an arc is subject to these transformations. In the case of an anisotropic mapping or transformation the rendered width of the line will change with the direction of the line segment. If the line width is specified as a scale factor it is not affected by any transformations.

Page 17

Add the following before the first paragraph of 4.6.2.3:

The following discussion applies to version 1 metafiles.

Page 17

Sub-clause 4.6.2.3: at the end of the first paragraph change "is not standardized." to the following:

is not standardized for version 1 metafiles.

Page 17

Add the following at the end of 4.6.2.3:

In version 2 metafiles, marker clipping is controlled by the MARKER CLIPPING MODE element, which can have one of the following values: 'locus', 'shape' or 'locus then shape'. However, clipping applies only if the CLIP INDICATOR is 'on'.

For 'locus' clipping, the specifying points of each marker are clipped at the intersection with the clip rectangle before shape rendering is applied. The marker is only visible if its specifying point is within the clip rectangle. Hence, part of the shape of a marker may appear outside the clip rectangle providing its specifying point is within the clip rectangle.

For 'shape' clipping, the shape of the rendered marker symbols are clipped to the intersection with the clip rectangle; that is, nothing is drawn outside the clip rectangle. Portions of the marker symbol may appear inside the clip rectangle even if the marker's position is outside.

For 'locus then shape' clipping, the clipping is first applied to the specifying points of each marker, as with 'locus' clipping, and then subsequently the rendered shape of the markers are again clipped.

If the marker size is measured in VDC units, it is subject to the VDC-to-Device mapping (4.4.7) as well as to both segment and copy transformation (4.12.4.2 and 4.12.5). The shape of markers is never affected by transformations; for example, a circle used as a marker type shall always appear as a circle. Only the marker size may be transformed. For this purpose, conceptually, vectors with length equal to the marker size and arbitrary orientations are transformed; the resulting marker size is determined by the orientation of the vector which maximizes the length under the transformation.

If the marker size is specified as a scale factor it is not affected by any transformations.

Page 18

Add the following at the end of 4.6.3.3:

Clipping of text strings is described in 4.7.6.

The vectors specified by the CHARACTER ORIENTATION element (4.7.6) are subject to the VDC-to-Device mapping (4.4.7) as well as to both segment and copy transformation (4.12.4.2 and 4.12.5).

Page 19

Add the following at the end of 4.6.4.5:

Edge clipping is controlled by the EDGE CLIPPING MODE element, which has the same enumerations as LINE CLIPPING MODE. Edges are clipped in the same way that lines are clipped; see 4.6.1.3.

Page 19

Add the following after 4.6.4.5:

4.6.4.6 Transformation

The entire mathematical locus of rectangles, circular and elliptical filled-area elements is subject to the VDC-to-Device mapping (4.4.7), segment transformations (2.12.4.2) and copy transformations (4.12.5). Because anisotropic transformation does not preserve angles between non-parallel lines, rectangles may become parallelograms and circles may become ellipses.

The vectors of the PATTERN SIZE element are subject to all transformations.

The edge widths are treated in exactly the same way as line widths (4.6.1.3).

Under certain conditions the clip rectangle is subject to the copy transformation (4.12.5).

Page 20

Add the following after 4.6.7

4.6.8 Closed figures

4.6.8.1 Construction of closed figures

A closed figure is a fill type compound object which commences with a BEGIN FIGURE element, followed by an ordered sequence of line and fill primitives (and optionally attributes and NEW REGION elements), and followed by END FIGURE. Edge attribute values are associated with the edge portions of the closed figure and fill attribute values are associated with the complete graphic object. BEGIN FIGURE and END FIGURE elements are delimiter elements; NEW REGION is a control element. The entire fill object is considered as a single unit on interpretation.

4.6.8.1.1 Closure point

The first point of the first line primitive in a new region is the closure point for that region. On interpretation this closure point is retained for use in closing the region. When the region is closed (with a NEW REGION or END FIGURE element, or by a fill primitive which begins a new region) an implicit boundary portion from the last point of the last line primitive in the region to this closure point is added to the closed figure on interpretation, unless these points are already coincident.

4.6.8.1.2 Regions

A closed figure consists of one or more regions. A region has a closed boundary which may be concave, convex, or self intersecting. A region is formed either by invoking a fill primitive inbetween BEGIN FIGURE and END FIGURE elements (FIGURE OPEN state; see 4.10) which closes the last region and contributes one or more complete regions, by invoking NEW REGION to start new regions to be formed from line primitives, or by a final invocation of END FIGURE. A closed figure constructed from only line primitives without use of NEW REGION consists of a single region.

The NEW REGION element may occur at any time during the closed figure construction. If the current region is closed, the element is ignored on interpretation. If the current region is open, an implicit boundary portion is added from the last point of the last primitive to the current closure point unless CONNECTING EDGE has been invoked after the last line primitive, in which case, an explicit boundary portion and edge portion is added by the CONNECTING EDGE line primitive.

4.6.8.2 Boundaries and edges

The boundary of each region consists of a combination of implicit boundary portions and edge portions.

4.6.8.2.1 Explicit boundary portions

Explicit boundary portions and edge portions are those added by the inclusion of primitives during closed figure construction. These are generated in the following situations:

- For fill primitives other than POLYGON SET, the complete edge becomes an explicit boundary portion and edge portion in the closed figure.

- For line primitives, those portions which would be rendered outside closed figure construction become explicit boundary portions and edge portions. In particular for DISJOINT POLYLINE, only the segments from the first point to the second point, from the third point to the fourth point, and so on, become explicit boundary portions and edge portions when incorporated into closed figures.

- A CONNECTING EDGE primitive which precedes an action which would normally have added an implicit boundary portion to the closed figure either to close a region (including closing the closed figure itself) or to connect two line primitives results in the portion added being an explicit boundary portion and edge portion. CONNECTING EDGE preceding or following DISJOINT POLYLINE or POLYGON SET does not affect the interpretation of those elements with respect to boundaries and edges.

Edge portions have associated edge attribute values taken from the current attribute values on interpretation. These values can be changed between the line and fill primitives that result in edge portions in a closed figure, and hence each edge portion has a distinct set of attribute values associated with it.

4.6.8.2.2 Implicit boundary portions

Edge attributes are never associated with implicit boundary portions. Implicit boundary portions are only rendered on interpretation for interior style HOLLOW and are a special representation of the interior, not a representation of any portion of the edge.

Implicit boundary portions are added on interpretation to the closed figure definition under the following circumstances:

- When NEW REGION, END FIGURE, or a fill primitive is interpreted and the current region has not been explicitly closed and CONNECTING EDGE has not occurred since the last line primitive, an implicit boundary portion is added from the last point of the last primitive to the current closure point to close the region.

- When the last point of the preceding line primitive is not coincident with the first point of the current line primitive, an implicit boundary portion is created to connect the last point of the preceding line primitive to the first point of the current line primitive.

- When portions of a DISJOINT POLYLINE primitive would not normally be rendered (i.e. from the second point to the third point, from the fourth point to the fifth point, and so on), implicit boundary portions are added between these points. (These are additional to the ones which may be added to connect to a preceding or following line primitive or to effect region closure after the disjoint polyline.)

- The portions of a POLYGON SET primitive as described below.

4.6.8.2.3 Conditions under which no boundary or edge is added

No boundary or edge portion is ever created connecting two regions, regardless of how those regions were created or closed.

4.6.8.3 Contribution of primitive elements to the closed figure

4.6.8.3.1 Contribution of line elements to the closed figure

For line primitives, the 'first point' of a line primitive is connected to the 'last point' of the preceding line primitive, and the connecting implicit boundary portion becomes part of the boundary of the closed figure on interpretation. For each of the line primitives the first and last points are defined to be as follows:

POLYLINE p1, p2, ..., pn:
p1 is the first point; pn is the last point.

DISJOINT POLYLINE p1, p2, ..., pn:
p1 is the first point; pn is the last point.

CIRCULAR ARC 3 POINT p1, p2, p3:
p1 is the first point; p3 is the last point.

CIRCULAR ARC CENTRE:
CIRCULAR ARC CENTRE REVERSED:
The first point is the intersection of the circle with the ray (dx start, dy start) from the centre point (i.e. the clockwise end of the arc for CIRCULAR ARC CENTRE, the anti-clockwise end of the arc for CIRCULAR ARC CENTRE REVERSED); the last point is the intersection of the circle with the ray (dx end, dy end) from the centre point (i.e. the anti-clockwise end of the arc for CIRCULAR ARC CENTRE, the clockwise end of the arc for CIRCULAR ARC CENTRE REVERSED).

ELLIPTICAL ARC:
The first point is the intersection of the ellipse with the ray (dx start, dy start) from the centre point; the last point is the intersection of the ellipse with the ray (dx end, dy end) from the centre point.

GENERALIZED DRAWING PRIMITIVE:
For GDPs which generate line primitives, the first point is the first point of the point list; and the last point is the last point of the point list, as defined in the in the GDP registration and associated documentation.

CONNECTING EDGE:
If the region is open, the start point of the connecting edge is the last point of the last line primitive, and the end point of the connecting edge is either the first point of the following primitive or the current closure point as described above. If the connecting edge would be of zero length (i.e. if the two points it connects are coincident), the element is ignored on interpretation. The current modal values of the edge attributes are associated with any edge portion generated by this element.

If the current region is not open, invocations of the CONNECTING EDGE elements encoutered are ignored on interpretation (i.e. CONNECTING EDGE shall not be used to connect regions).

Invoking CONNECTING EDGE multiple times after a line primitive results in the first instance (with its associated attributes) being used on interpretation.

On interpretation the theoretical definitions of the line primitives, not their renditions on the display surface, are used to define the explicit boundary portions of the closed figure. In particular, clipping does not apply to the construction of the closed figure, and the gaps or spaces of the edge type or the rendered width of the edge width do not affect the definition of the boundary of the closed figure.

4.6.8.3.2 Contribution of fill elements to the closed figure

Each fill primitive contributes a complete region to the figure (POLYGON SET may contribute more than one), after first closing the current region if one is open. On interpretation, an implicit NEW REGION is performed before and after a fill primitive (i.e. the new region resulting from a fill primitive is closed, and the next primitive begins a new region.)

The unclipped boundary of each fill primitive contributes to the unclipped boundary of the closed figure.

POLYGON SET primitives contribute to closed figure construction as follows:

- A POLYGON SET is considered to contribute one or more complete regions. If the current region has not been closed, an implicit NEW REGION is performed before the POLYGON SET is added to the figure definition. If the POLYGON SET does not end with a point whose edge-out flag is 'close visible' or 'close invisible', an implicit NEW REGION is performed after the POLYGON SET.
- Sequences of points with edge-out flag 'visible' are treated as if they were polylines, terminating with the first point with a different edge-out flag. Each such polyline becomes an edge portion of the boundary of the figure. The edge attribute values (including EDGE VISIBILITY) in effect when POLYGON SET occurs are associated on interpretation with any edge portion added in this way.
- Sequences of points with edge-out flag 'invisible' contribute implicit boundary portions which are polylines joining the points in the sequence, but not edges. Edge attribute values are not associated with these.
- Points with edge-out flag 'close invisible' generate the equivalent of a NEW REGION, generating an implicit boundary portion from this point to the current closure point if these are not coincident, and closing the current region.
- Points with edge-out flag 'close visible' generate the equivalent of a CONNECTING EDGE followed by a NEW REGION, resulting in an edge portion from this point to the current closure point if these are not coincident. The edge attribute values (including EDGE VISIBILITY) in effect when POLYGON SET is invoked are associated with any edge portion added in this way.

4.6.8.3.3 Contribution of GDPs to the closed figure

A GDP which is defined as a line primitive shall specify which is the first point and the last point in its point list, with respect to closed figure construction. Such GDPs are assumed to contribute to a closed figure a boundary corresponding to the unclipped locus which would be rendered on interpretation if the element occurred outside closed figure construction. Any other behaviour shall be as documented explicitly in the GDP description. A GDP which is defined as being a fill primitive is treated as described in the previous section. Any variation or special handling for closed figure construction shall be documented explicitly in the GDP description.

4.6.8.5 Examples of closed figures

Examples of closed figures are shown in Figure 2a, 2b, 2c, 2d, 2e and 2f.

The POLYGON SET example shown in Figure 13 may also be obtained using the closed figure:

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
        POLYLINE (P3, P1, P2)
        NEW REGION {Note 1}
        POLYLINE (P4, P5, P6, P4)
END FIGURE
```

NOTE

1 Invisible implicit boundary portion P3..P2 generated.

Figure 2a shows the closed figure resulting from interpretation of the elements listed below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
        POLYLINE (P1, P2)
        CIRCULAR ARC 3 POINT (P2, P3, P4)
        POLYLINE (P4, P5)
        CIRCULAR ARC 3 POINT (P5, P6, P1)
END FIGURE
```

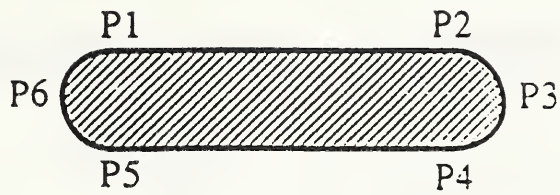


Figure 2a

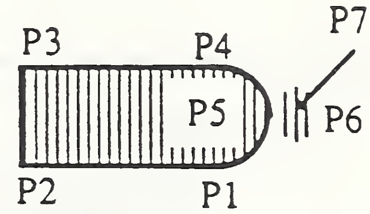


Figure 2b

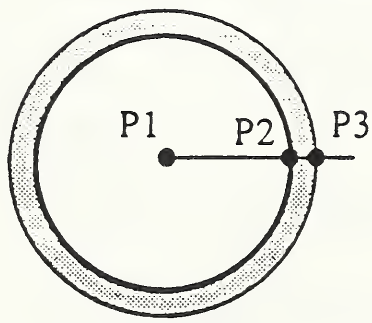


Figure 2c

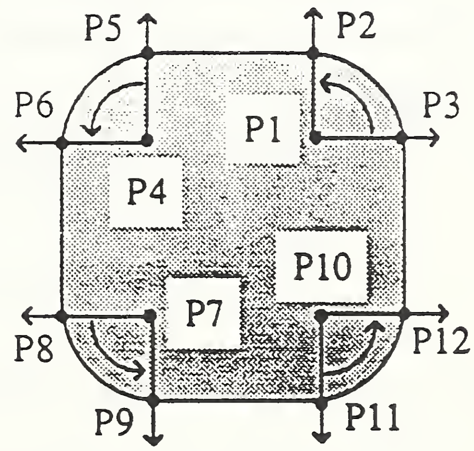


Figure 2d

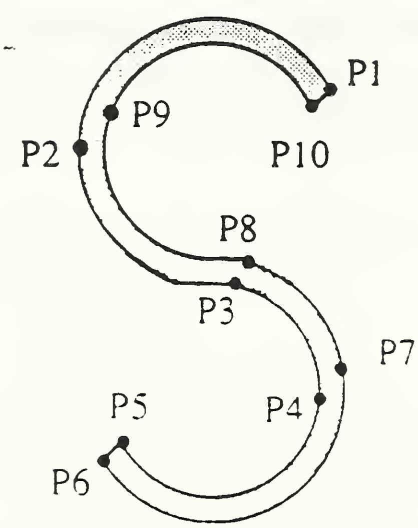


Figure 2e

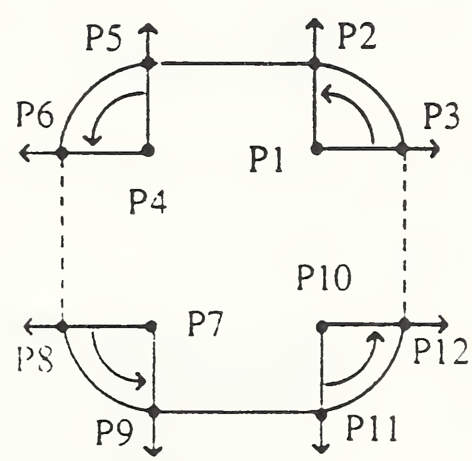


Figure 2f

Figures 2a, 2b, 2c, 2d, 2e, 2f - Examples of closed figures

Figure 2a could also be the result of interpreting the following sequence of elements which include CONNECTING EDGE.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
    CIRCULAR ARC 3 POINT (P2, P3, P4)
    CONNECTING EDGE
    CIRCULAR ARC 3 POINT (P5, P6, P1) {Note 1}
    CONNECTING EDGE
END FIGURE {Note 2}
```

NOTES

1 Visible edge portion P4..P5 generated.
2 Visible edge portion P1..P2 generated.

Figure 2b shows the closed figure resulting from interpretation of the elements listed below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
    POLYLINE (P1, P2, P3, P4)
    CIRCULAR ARC 3 POINT (P4, P5, P1)
    EDGE VISIBILITY (OFF)
    NEW REGION
    P7 = P5 + (P6-P5)/2
    CIRCULAR ARC CENTRE (P7, 1, 0, 1, 0, |P7 - P5|)
END FIGURE
```

Figure 2c shows the closed figure resulting from interpretation of the elements listed below.

```
BEGIN FIGURE
    CIRCULAR ARC CENTRE (P1, 1, 0, 1, 0, |P3 - P1|)
    NEW REGION
    CIRCULAR CENTRE (P1, 1, 0, 1, 0, |P2 - P1|)
END FIGURE
```

Figure 2c could also be the result of interpreting the following sequence of elements which include fill area elements.

```
BEGIN FIGURE
    CIRCLE (P1, |P3 - P1|)
    CIRCLE (P1, |P2 - P1|)
END FIGURE
```

Figure 2d shows the use of ELLIPTICAL ARC to draw a box with rounded corners and is the result of interpreting the sequence of elements shown below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
    ELLIPTICAL ARC (P1, P2, P3, (1,0), (0,1))
    CONNECTING EDGE
    ELLIPTICAL ARC (P4, P5, P6, (0,1), (-1,0)) {Note 1}
    CONNECTING EDGE
    ELLIPTICAL ARC (P7, P8, P9, (-1,0), (0,-1))
    CONNECTING EDGE
    ELLIPTICAL ARC (P10, P11, P12, (0,-1), (1,0))
    CONNECTING EDGE
END FIGURE {Note 2}
```

NOTES

1 Visible edge portion P2..P5 generated; edge portions P6..P8 and P9..P11 are drawn with the next two arcs.
2 Visible edge portion P12..P3 generated.

Figure 2e shows the use of CIRCULAR ARC 3 POINT to create an 'S' shape and is the result of interpreting the sequence of elements shown below.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
        CIRCULAR ARC 3 POINT (P1, P2, P3)
        CIRCULAR ARC 3 POINT (P3, P4, P5)
        CONNECTING EDGE
        CIRCULAR ARC 3 POINT (P6, P7, P8) {Note 1}
        CIRCULAR ARC 3 POINT (P8, P9, P10)
        CONNECTING EDGE
END FIGURE {Note 2}
```

NOTES

1 Visible edge portion P5..P6 generated.
2 Visible edge portion P10.P1 generated.

Figure 2f shows the closed figure resulting from interpretation of the elements listed below. It is similar to figure 2d, but makes use of changing the edge attributes between successive occurrences of CONNECTING EDGE.

```
EDGE VISIBILITY (ON)
BEGIN FIGURE
        ELLIPTICAL ARC(P1, P2, P3, (1,0), (0,1))
        ELLIPTICAL ARC(P4, P5, P6, (0,1), (-1,0)) {Note 1}
        EDGE TYPE (DASHED)
        CONNECTING EDGE
        ELLIPTICAL ARC(P7, P8, P9, (-1,0), (0,-1)) {Note 2}
        EDGE TYPE (SOLID)
        CONNECTING EDGE
        ELLIPTICAL ARC(P10, P11, P12, (0,-1), (1,0))
        EDGE TYPE(DASHED)
        CONNECTING EDGE
END FIGURE {Note 3}
```

NOTES

1 No edge portion P2..P5 generated.
2 Visible (dashed) edge portion P6..P8 generated; solid edge portion P9..P11 drawn with the next arc.
3 Visible (dashed) edge portion P12..P3 generated.

Page 39

Add the following after 4.7.8:

4.7.9 Pick identifier

The pick identifier is associated with graphical primitive elements within segments (see 4.12). It is the only attribute element which does not affect the appearance of a graphical primitive element. It merely establishes a means of identification of primitives within segments at metafile interpretation. The PICK IDENTIFIER element has no graphical effect.

Page 40

Add the following after 4.11:

4.12 Segment elements

4.12.1 Introduction

In the CGM graphic objects may be grouped in segments, each segment being identified by a unique segment identifier. Segments may have the attributes:

a) transformation;

b) highlighting;

c) display and pick priority.

These may be defined at segment definition time, before the first primitives of the segment, and shall not be changed thereafter.

Only elements inside segments are affected by the segment attributes.

The segment elements are:

COPY SEGMENT
INHERITANCE FILTER
CLIP INHERITANCE
SEGMENT TRANSFORMATION
SEGMENT HIGHLIGHTING
SEGMENT DISPLAY PRIORITY
SEGMENT PICK PRIORITY

Segments are delimited by BEGIN SEGMENT and END SEGMENT.

4.12.2 Local and global segments

There are two types of segments: local segments and global segments. Both contain primitives and attributes that can be manipulated in the manner described above. Local segments have no existence beyond the bounds of the picture body in which they are defined. Defining a local segment in a picture automatically includes that segment in the picture's image. In contrast, global segments can be referenced by any of the pictures in the metafile in which they are defined.

4.12.2.1 Location of, and access to, global segments.

A global segment is delimited by the BEGIN SEGMENT and END SEGMENT elements. Global segments are defined in the Metafile Descriptor. They are not a part of any picture within the metafile. They shall be accessed from within individual pictures by the COPY SEGMENT (4.12.5) element. The COPY SEGMENT element incorporates the segment into the open picture in the same way for both local and global segments.

4.12.2.2 Permitted segment-related elements in the Metafile Descriptor

BEGIN SEGMENT is the only segment-related element that is allowed within the Metafile Descriptor State (MDS) (see 4.10). BEGIN SEGMENT changes the state to Global Segment State (GSS).

4.12.2.3 References to global segments

Within pictures, no elements are allowed that would modify the contents or default appearance of global segments. This restriction preserves the logical independence of pictures and the ability to randomly access pictures. The only references to global segments within pictures shall be by using the COPY SEGMENT element.

4.12.2.4 Association of control and attribute elements with primitives inside segments

The current modal values of control and attribute elements are associated with the primitives inside local segments. The modal values established by setting control or attribute elements within a segment remain in effect outside the segment until they are explicitly changed.

Control and attribute elements are bound in global segments as they are in local segments. Upon the occurrence of BEGIN METAFILE, every element that is modally defined and bound to primitives (Metafile Descriptor elements defining modes and precisions, Picture Descriptor elements, Control elements, Attribute elements and Segment Control elements) has a default value. Conceptually the set of all of these define a "modal state list".

The Metafile Descriptor (MD) is processed sequentially. Throughout the Metafile Descriptor, modal MD elements modify the MD entries in the state list and occurrences (possibly multiple) of the METAFILE DEFAULTS REPLACEMENT element allow manipulation (outside of GSS state) of the rest of the modal elements (as well as explicitly changing the defaults). Within GSS state the allowable modal (control, attribute, and segment attribute) elements also alter the contents of the modal state list. The values of modal elements that are in effect upon BEGIN PICTURE are the default values for that picture, whether they are implicit (defined in ISO/IEC 8632) or explicit (that is, by values set in the Metafile Defaults Replacement).

4.12.3 Delimiting and naming segments

The contents of a segment are delimited by the elements BEGIN SEGMENT and END SEGMENT which are delimiter elements. The elements in between these two delimiters are a part of that segment. Each segment has an identifier associated with it. No two global segments shall have the same identifier and no local segment shall have an identifier which is the same as either a local segment in the same picture or the same as a global segment.

4.12.4 Segment attributes

4.12.4.1 Introduction

The segment attributes associated with each segment control its display. Segment attributes shall be set only after the segment has been opened with the BEGIN SEGMENT element. When a segment is opened the segment's attributes are set to their default values. Segment attributes, if set, shall be set immediately after the BEGIN SEGMENT element and before any other type of element. This structure is shown below:

BEGIN SEGMENT (Segment identifier)
Segment attributes
Allowed primitives, attributes and control elements in any order
END SEGMENT

4.12.4.2 Segment transformation

The segment transformation is a coordinate transformation associated with each segment and applies to all graphical objects in the identified segment and will be used on interpretation. Clipping rectangles are not transformed by the segment transformation. It allows scaling, translation, and rotation of segments to be defined during segment definition.

The segment transformation is a transformation of VDC space to VDC space and is distinct from the VDC-to-Device mapping which is a transformation of VDC space to device coordinate space.

The transformation attribute of a segment may be defined by the SEGMENT TRANSFORMATION element during the segment definition. A segment transformation is represented by a 2 x 3 matrix, comprising a 2 x 2 scaling and rotation portion, and a 2 x 1 translation portion. If the SEGMENT TRANSFORMATION element is not stored in the metafile, then all coordinate data is mapped using only the VDC-to-Device mapping. If the SEGMENT TRANSFORMATION is stored in the metafile, it is applied before the application of the VDC-to-Device mapping.

The use of segment transformations may produce coordinates that cannot be expressed within the VDC range. This is handled in an interpretation dependent way.

4.12.4.3 Segment highlighting

Segment highlighting can take one of two values, NORMAL or HIGHLIGHTED. The setting of this attribute selects one of these two states for the segment.

4.12.4.4 Segment display priority

The display priority attribute of a segment determines how overlapping segments are displayed. During interpretation segments with higher display priorities will be displayed as if they were in front of segments with lower display priorities. The segment display priority may be normalized to the continuous range of real numbers, zero to one, by applying the minimum extent and maximum extent values provided by the Metafile Descriptor element SEGMENT PRIORITY EXTENT.

4.12.4.5 Segment pick priority

The pick priority attribute of a segment is used to resolve the picking of segments which overlap. The segment pick priority may be normalized to the continuous range of real numbers, zero to one, by applying the minimum extent and maximum extent values provided by the Metafile Descriptor element SEGMENT PRIORITY EXTENT. Interpretation of SEGMENT PICK PRIORITY has no graphical effect.

4.12.5 Copy segment and inheritance

The COPY SEGMENT element inserts the elements of the referenced segment into the picture at the point of occurrence of the element.

The elements copied may be altered in a variety of ways:

a) The inheritance filter mechanism controls whether individual attribute values are reapplied to the elements.

b) The clip inheritance mechanism controls whether the primitives in the segment are clipped to the current clip rectangle or to a combination of the current and the segment clipping rectangles.

c) The primitive elements are transformed by the copy transformation and optionally by the segment transformation of the copied segment according to the rules for transformation.

COPY SEGMENT has a transformation matrix as a parameter. The copy transformation is applied to graphical objects before they are copied. This also applies to clipping rectangles in the segment (see below). Graphical objects may be transformed to alter their location, size, and orientation.

A segment may be referenced by the COPY SEGMENT element, either within a picture or in a global segment. The attributes associated on interpretation can be those bound to the segment being copied or can be imposed by the inclusion of the INHERITANCE FILTER element.

The clipping associated with a segment can be that associated with the picture at the time of the copy or can be a combination of the current clipping and the segment clipping when the CLIP INHERITANCE element is used.

The inheritance filter mechanism allows the use of the current values of attributes and controls to be associated with the copied segment in place of the attributes and controls bound to the primitives when the segment was created. The attributes and controls to be associated with the segment can be all attributes or can be a subset of attributes. The attributes and controls are selected using the INHERITANCE FILTER element. The attributes and controls can be selected using individual or group names for attributes, controls and ASFs. The elements that can be selected are shown in table 3a for attributes and controls (both individual element names and group names) and in table 3b for ASFs.

If an attribute or group of attributes designated in the filter selection list is set to 'state list', graphic objects inherit that attribute or group of attributes from the current modal values when a segment is copied.

If an attribute or group of attributes designated in the filter selection list is set to 'segment', that attribute or group of attributes is unaffected (in all graphic objects employing them) by the corresponding current state list when a segment is copied.

The default inheritance filter setting value is 'segment' for all attributes and controls.

Table 3a - Inheritance filter selection names for attributes

Attribute Group Name	Individual Attribute Name
LINE ATTRIBUTES	LINE BUNDLE INDEX
	LINE TYPE
	LINE WIDTH
	LINE COLOUR
	LINE CLIPPING MODE
MARKER ATTRIBUTES	MARKER BUNDLE INDEX
	MARKER TYPE
	MARKER SIZE
	MARKER COLOUR
	MARKER CLIPPING MODE
TEXT PRESENTATION AND PLACEMENT ATTRIBUTES	TEXT BUNDLE INDEX
	TEXT FONT INDEX
	TEXT PRECISION
	CHARACTER EXPANSION FACTOR
	CHARACTER SPACING
	TEXT COLOUR
TEXT PLACEMENT AND ORIENTATION ATTRIBUTES	CHARACTER HEIGHT
	CHARACTER ORIENTATION
	TEXT PATH
	TEXT ALIGNMENT
FILL ATTRIBUTES	FILL BUNDLE INDEX
	INTERIOR STYLE
	FILL COLOUR
	HATCH INDEX
	PATTERN INDEX
EDGE ATTRIBUTES	EDGE BUNDLE INDEX
	EDGE TYPE
	EDGE WIDTH
	EDGE COLOUR
	EDGE VISIBILITY
	EDGE CLIPPING MODE
PATTERN ATTRIBUTES	FILL REFERENCE POINT
	PATTERN SIZE
OUTPUT CONTROL	AUXILIARY COLOUR
	TRANSPARENCY
PICK IDENTIFIER	PICK IDENTIFIER
ALL ATTRIBUTES AND CONTROL	All attributes and control elements
ALL	All attributes, control elements and ASFs

Table 3b - Inheritance filter selection names for Aspect Source Flags

ASF Group Name	Individual ASF Name
LINE ASFS	LINE TYPE ASF
	LINE WIDTH ASF
	LINE COLOUR ASF
MARKER ASFS	MARKER TYPE ASF
	MARKER SIZE ASF
	MARKER COLOUR ASF
TEXT ASFS	TEXT FONT INDEX ASF
	TEXT PRECISION ASF
	CHARACTER EXPANSION FACTOR ASF
	CHARACTER SPACING ASF
	TEXT COLOUR ASF
FILL ASFS	INTERIOR STYLE ASF
	FILL COLOUR ASF
	HATCH INDEX ASF
	PATTERN INDEX ASF
EDGE ASFS	EDGE TYPE ASF
	EDGE WIDTH ASF
	EDGE COLOUR ASF
ALL ASFS	All aspect source flags

An example of the COPY SEGMENT element with the INHERITANCE FILTER element is as follows:

```
BEGIN METAFILE "..."
  .
  .
BEGIN SEGMENT (1)
        LINE COLOUR (blue)
        POLYLINE                                                   blue solid line
END SEGMENT

BEGIN DEFAULTS REPLACEMENT
        LINE TYPE (dash)
END DEFAULTS REPLACEMENT

BEGIN SEGMENT (2)
        LINE COLOUR (red)
        INHHERITANCE FILTER (LINE ATTRIBUTES,STATE LIST)
        COPY SEGMENT (1)                                           red dashed line
        POLYLINE                                                   red dashed line
        INHERITANCE FILTER (LINE ATTRIBUTES,SEGMENT)
        COPY SEGMENT (1)                                           blue solid line
        POLYLINE                                                   red dashed line
END SEGMENT

BEGIN PICTURE "..."
BEGIN PICTURE BODY
LINE COLOUR (green)
INHERITANCE FILTER (LINE ATTRIBUTES,SEGMENT)
COPY SEGMENT (2)                                                   red dashed line
                                                                   red dashed line
                                                                   blue solid line
                                                                   red dashed line
POLYLINE                                                           green dashed line
INHERITANCE FILTER (LINE ATTRIBUTES,STATE LIST)
COPY SEGMENT (2)                                                   green dashed line
                                                                   green dashed line
                                                                   green dashed line
                                                                   green dashed line
BEGIN SEGMENT (3)
        LINE COLOUR (red)
        COPY SEGMENT (1)                                           red dashed line
        INHERITANCE FILTER (LINE ATTRIBUTES,SEGMENT)
        COPY SEGMENT (1)                                           blue solid line
END SEGMENT
  .
  .
LINE COLOUR (green)
COPY SEGMENT (3)                                                   red dashed line
                                                                   blue solid line
INHERITANCE FILTER (LINE ATTRIBUTES,STATE LIST)
COPY SEGMENT (3)                                                   green dashed line
                                                                   green dashed line
  .
  .
  .
END PICTURE
END METAFILE
```

Clipping is not included in the INHERITANCE FILTER. There is a separate element that controls clipping behaviour - CLIP INHERITANCE. Its values may be either 'state list' or 'intersection'.

If the value is 'state list', then the clip rectangle associated with primitives in the copied segment is that of the last CLIP RECTANGLE encountered during interpretation in the metafile element sequence prior to the COPY SEGMENT element, that is, the value in the "modal state list".

If the value is 'intersection' and if both the modal state list clip indicator and the clip indicator associated with the primitives of the copied segment are 'on', then the resulting clipping boundary is the intersection of the modal state list clip rectangle with the clipping boundary resulting from the application of the copy transformation to the clip rectangle associated with the primitives. If either indicator is 'off', then there is no contribution from its associated clip rectangle. To illustrate: if TA and TB are copy transformations:

```
BEGIN SEGMENT A
CLIP INDICATOR(ON)
CLIP RECTANGLE R1
POLYLINE P1
END SEGMENT

CLIP INHERITANCE (INTERSECTION)
CLIP INDICATOR(ON)
CLIP RECTANGLE R2
POLYLINE P2
COPY SEGMENT (A,TA)
POLYLINE P3
```

P2 and P3 are clipped by R2, P1 is clipped by R2 (intersected with) TA(R1). This clipping region may turn out to be an 8-sided convex polygon, if TA causes rotation and skewing.

The composition of clipping rectangles continues however many levels the segment hierarchy is nested. For example:

```
BEGIN SEGMENT A
CLIP RECTANGLE R0
POLYLINE P0
CLIP RECTANGLE  R1
POLYLINE P1
END SEGMENT

BEGIN SEGMENT B
CLIP RECTANGLE R2
POLYLINE P2
CLIP INHERITANCE (INTERSECTION)
COPY SEGMENT (A,TA)
END SEGMENT

CLIP RECTANGLE R3
CLIP INHERITANCE (INTERSECTION)
COPY SEGMENT (B,TB)
POLYLINE P3
```

The effective clipping "rectangles" are:

for P0: TB(R2 intersection TA(R0)) intersection R3
for P1: TB(R2 intersection TA(R1)) intersection R3
for P2: TB(R2) intersection R3
for P3: R3

From this example it can be seen that the effective clipping "rectangle" can in fact be an arbitrary convex polygon. Annex D contains recommended fallback procedures for interpreters which cannot perform such clipping.

Segment Transformations are never applied to clipping boundaries. The default value for CLIP INHERITANCE is 'state list'.

Page 39

Sub-clause 4.10: Change the text in the third paragraph, sixth line from "figure 12" to:

figures 12 and 12a

Page 40

Add the following at the end of 4.10:

The states in which each element is allowed for version 2 metafiles are also described in table 3c.

Page 41

Change the title of Figure 12 to be:

Figure 12 - State diagram for version 1 metafiles

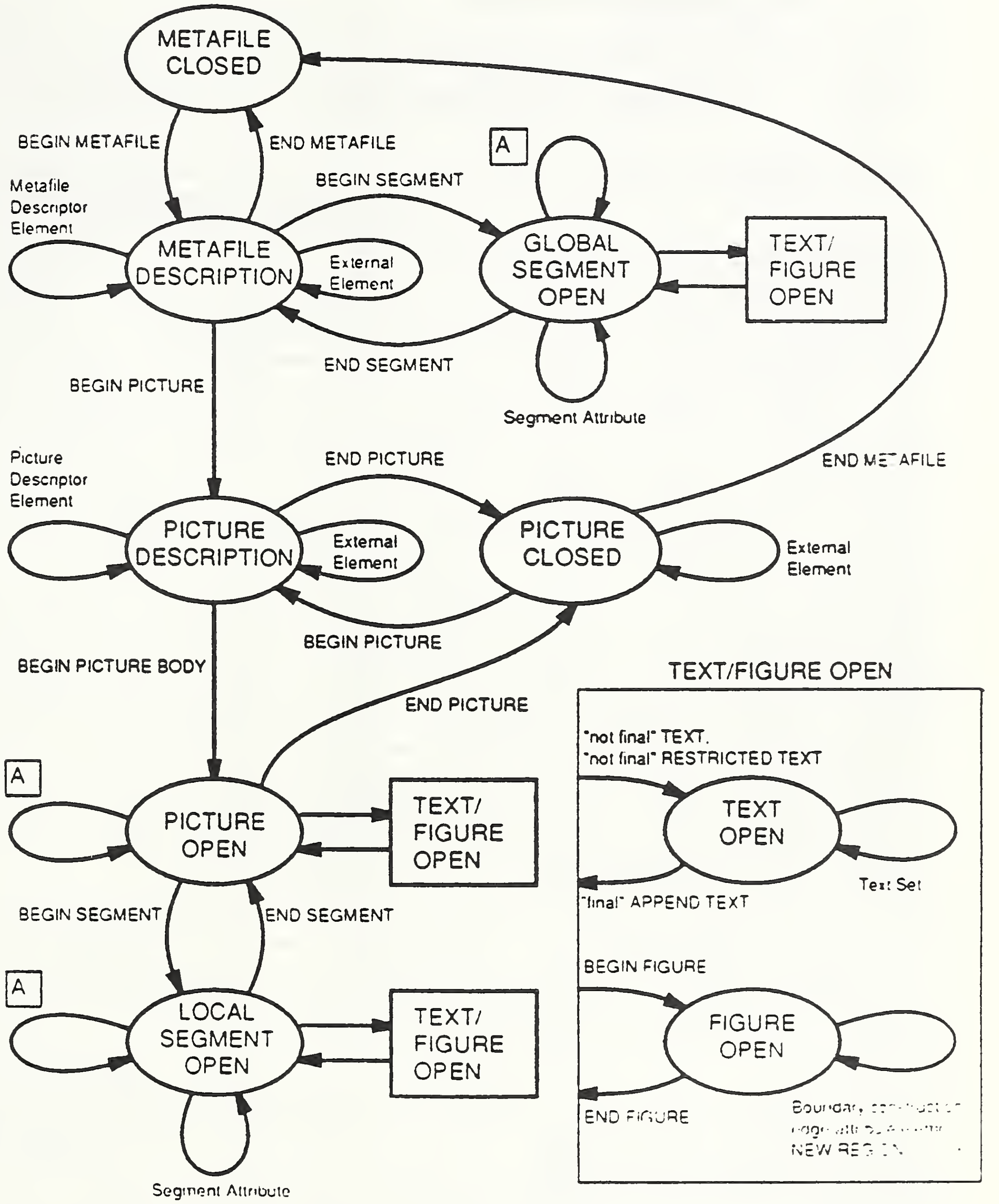
Page 41

Add the following text after the state diagram

NOTE - Many elements allowed in state PO can also occur in the METAFILE DEFAULTS REPLACEMENT.

Page 41

Add figure 12a



A All Control, Graphical Primitive, Attribute, Escape and External Elements except NEW REGION, CONNECTING EDGE, APPEND TEXT, "not final" TEXT, "not final" RESTRICTED TEXT plus COPY SEGMENT, INHERITANCE FILTER and CLIP CONTROL

Text Set: "not final" APPEND TEXT, TEXT BUNDLE INDEX, TEXT FONT INDEX, TEXT PRECISION, CHARACTER EXPANSION FACTOR, CHARACTER SPACING, TEXT COLOUR, CHARACTER HEIGHT, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX, [illegible]

Figure 12a - State diagram for version 2 metafiles

Page 41

Add the following table after Figure 12:

Table 3c - CGM Elements by their allowed states

CGM Element	CGM States								
	PCS	MDS	DR (4)	GSS	PDS	POS	TOS	LSS	FOS
BEGIN METAFILE (1)									
BEGIN PICTURE	X	X							
BEGIN PICTURE BODY					X				
END PICTURE						X			
BEGIN SEGMENT		X				X			
END SEGMENT				X				X	
BEGIN FIGURE				X		X		X	
END FIGURE									X
END METAFILE	X								
METAFILE VERSION		X							
METAFILE DESCRIPTION		X							
VDC TYPE		X							
INTEGER PRECISION		X							
REAL PRECISION		X							
INDEX PRECISION		X							
COLOUR PRECISION		X							
COLOUR INDEX PRECISION		X							
NAME PRECISION		X							
MAXIMUM COLOUR INDEX		X							
COLOUR VALUE EXTENT		X							
METAFILE ELEMENT LIST		X							
METAFILE DEFAULTS REPLACEMENT		X							
FONT LIST		X							
CHARACTER SET LIST		X							
CHAR CODING ANNOUNCER		X							
METAFILE CATEGORY		X							
MAXIMUM VDC EXTENT		X							
SEGMENT PRIORITY EXTENT		X							
SCALING MODE			X		X				
COLOUR SELECTION MODE			X	X	X	X		X	
LINE WIDTH SPECIFICATION MODE			X	X	X	X		X	
MARKER SIZE SPECIFICATION MODE			X	X	X	X		X	
EDGE WIDTH SPECIFICATION MODE			X	X	X	X		X	
VDC EXTENT			X		X				
BACKGROUND COLOUR			X		X				
DEVICE VIEWPORT			X		X				
DEVICE VIEWPORT MAPPING			X		X				
DEVICE VIEWPORT SPECIFICATION MODE			X		X				

Table 3c (continued)

CGM Element	CGM States								
	PCS	MDS	DR	GSS	PDS	POS	TOS	LSS	FOS
LINE REPRESENTATION			X		X				
MARKER REPRESENTATION			X		X				
TEXT REPRESENTATION			X		X				
FILL REPRESENTATION			X		X				
EDGE REPRESENTATION			X		X				
VDC INTEGER PRECISION			X	X		X		X	X
VDC REAL PRECISION			X	X		X		X	X
AUXILIARY COLOUR			X	X		X	X	X	X
TRANSPARENCY			X	X		X	X	X	X
CLIP RECTANGLE			X	X		X		X	
CLIP INDICATOR			X	X		X		X	
LINE CLIPPING MODE			X	X		X		X	
MARKER CLIPPING MODE			X	X		X		X	
EDGE CLIPPING MODE			X	X		X		X	
NEW REGION									X
SAVE PRIMITIVE CONTEXT				X		X		X	
RESTORE PRIMITIVE CONTEXT				X		X		X	
POLYLINE				X		X		X	X
DISJOINT POLYLINE				X		X		X	X
POLYMARKER				X		X		X	
TEXT				X		X		X	
RESTRICTED TEXT				X		X		X	
APPEND TEXT							X		
POLYGON				X		X		X	X
POLYGON SET				X		X		X	X
CELL ARRAY				X		X		X	
GDP				X		X		X	X
RECTANGLE				X		X		X	X
CIRCLE				X		X		X	X
CIRCULAR ARC 3 POINT				X		X		X	X
CIRCULAR ARC 3 POINT CLOSE				X		X		X	X
CIRCULAR ARC CENTRE				X		X		X	X
CIRCULAR ARC CENTRE CLOSE				X		X		X	X
ELLIPSE				X		X		X	X
ELLIPTICAL ARC				X		X		X	X
ELLIPTICAL ARC CLOSE				X		X		X	X
CIRCULAR ARC CENTRE REVERSED				X		X		X	X
CONNECTING EDGE									X
LINE BUNDLE INDEX			X	X		X		X	
LINE TYPE			X	X		X		X	
LINE WIDTH			X	X		X		X	
LINE COLOUR			X	X		X		X	

Table 3c (concluded)

CGM Element	CGM States								
	PCS	MDS	DR	GSS	PDS	POS	TOS	LSS	FOS
MARKER BUNDLE INDEX			X	X		X		X	
MARKER TYPE			X	X		X		X	
MARKER SIZE			X	X		X		X	
MARKER COLOUR			X	X		X		X	
TEXT BUNDLE INDEX			X	X		X	X	X	
TEXT FONT INDEX			X	X		X	X	X	
TEXT PRECISION			X	X		X	X(3)	X	
CHARACTER EXPANSION FACTOR			X	X		X	X	X	
CHARACTER SPACING			X	X		X	X	X	
TEXT COLOUR			X	X		X	X	X	
CHARACTER HEIGHT			X	X		X	X	X	
CHARACTER ORIENTATION			X	X		X		X	
TEXT PATH			X	X		X		X	
TEXT ALIGNMENT			X	X		X		X	
CHARACTER SET INDEX			X	X		X	X	X	
ALTERNATE CHARACTER SET INDEX			X	X		X	X	X	
FILL BUNDLE INDEX			X	X		X		X	
INTERIOR STYLE			X	X		X		X	
FILL COLOUR			X	X		X		X	
HATCH INDEX			X	X		X		X	
PATTERN INDEX			X	X		X		X	
EDGE BUNDLE INDEX			X	X		X		X	X
EDGE TYPE			X	X		X		X	X
EDGE WIDTH			X	X		X		X	X
EDGE COLOUR			X	X		X		X	X
EDGE VISIBILITY			X	X		X		X	X
FILL REFERENCE POINT			X	X		X		X	
PATTERN TABLE			X		X	X			
PATTERN SIZE			X	X		X		X	
COLOUR TABLE			X		X	X			
ASPECT SOURCE FLAGS			X	X		X		X	X(2)
PICK IDENTIFIER			X	X		X		X	
ESCAPE	X	X	X	X	X	X	X	X	X
MESSAGE	X	X	X	X	X	X		X	X
APPLICATION DATA	X	X	X	X	X	X		X	X
COPY SEGMENT				X		X		X	
INHERITANCE FILTER			X	X		X		X	
CLIP INHERITANCE			X	X		X		X	
SEGMENT TRANSFORMATION				X				X	
SEGMENT HIGHLIGHTING				X				X	
SEGMENT DISPLAY PRIORITY				X				X	
SEGMENT PICK PRIORITY				X				X	

Abbreviations used in table 3c:

PCS	Picture Closed State
MDS	Metafile Description State
DR	Defaults Replacement Mode
GSS	Global Segment State
PDS	Picture Description State
POS	Picture Open State
TOS	Text Open (Partial text) State
LSS	Local Segment State
FOS	Figure Open State

NOTES

1 BEGIN METAFILE is the only element allowed in the state 'Metafile Closed'
2 Edge ASFs are the only ASFs allowed in Figure Open State
3 Use of TEXT PRECISION in text open state is permitted; however, the intended result is not well defined and such usage is likely to lead to unpredictable results.
4 Defaults replacement mode is not actually a metafile state, but it is included in this table for the convenience of the implementor of this standard.

Page 42

Sub-clause 5.1: Add the following after the ninth paragraph which starts with the sentence: "The External Elements....":

The segment elements (see 5.10) provide for the grouping and manipulation of elements.

Page 42

Sub-clause 5.1: Add the following at the end of the table of abbreviations of data type names:

N	Name	Identifier for a segment, pick or context. Realization is integer: range is dependent on NAME PRECISION
VC	Viewport Coordinate	Single real or integer value as determined by the DEVICE VIEWPORT SPECIFICATION MODE: R fraction [0..1] of default viewport I millimetres (scaled) I native device units
VP	Viewport Point	Two VC values representing the x and y coordinates of a point in viewport specification space

Page 46

Add the following after 5.2.5:

5.2.6 BEGIN SEGMENT

Parameters:

segment identifier (N)

Description:

This is the first element of a segment. All subsequent elements until the next END SEGMENT will belong to this segment.

References:

4.2
4.12.3

5.2.7 END SEGMENT

Parameters:

None

Description:
Subsequent elements will no longer belong to a segment.

References:
4.2
4.12.3

5.2.8 BEGIN FIGURE

Parameters:

none

Description:
This is the first element of a closed figure. All subsequent elements until the next END FIGURE will be part of the closed figure.

Reference:
4.6.8

5.2.9 END FIGURE

Parameters:

none

Description:
This element terminates the current closed figure.

If the current region has not yet been closed by a preceding NEW REGION element and if the last point of the last line element is not coincident with the current closure point, then the current subregion is closed by a line segment connecting the last point of the preceding line element to the current closure point. This line becomes a part of the implicit boundary specification. If the END FIGURE was preceded by a CONNECTING EDGE element, which was itself preceded by a line primitive, then this line also becomes part of the edge specification. If the region which has been previously closed is empty, or if the last point of the last line element is coincident with the current closure point, or if the last element was a filled-area primitive, then no line segment is generated by this element.

Reference:
4.6.8

Page 47

Add the following at the end of the "Description" in 5.3.1:

The CGM as defined in ISO 8632 : 1987/Am.1 : 1990 is version two (2).

METAFILE VERSION shall appear only once in the Metafile Descriptor for version 2 metafiles. It is recommended that it shall appear only once in version 1 metafiles.

Page 50

Sub-clause 5.3.11: Add the following shorthand names at the end of the list given in the second paragraph of the "Description":

VERSION 2 SET
EXTENDED PRIMITIVES SET
VERSION 2 GKSM SET

Page 50

Sub-clause 5.3.11: Add the following at the end of the "Description":

METAFILE ELEMENT LIST shall appear only once in the Metafile Descriptor for version 2 metafiles. It is recommended that it shall appear only once in version 1 metafiles.

Page 55

Add the following sub-clauses after 5.3.15:

5.3.16 NAME PRECISION

Parameters:

The form of the parameter depends on the specific encoding.

Description:

The precision for operands of data type name (N) is specified for subsequent data of type N. The precision is defined as the field width measured in units applicable to the specific encoding.

Reference:

4.3

5.3.17 MAXIMUM VDC EXTENT

Parameters:

first corner (P)
second corner (P)

Description:

The two corners define a rectangular extent in VDC space which bounds the values of the VDC EXTENT elements which may be found in the metafile. It may be, but need not be, a closest bound in the sense that it exactly equals the union of the extent rectangles in the metafile.

References:

4.3
4.4.4

5.3.18 SEGMENT PRIORITY EXTENT

Parameters:

minimum priority extent (I)
maximum priority extent (I)

Description:

The parameters represent an extent which bounds the segment display and pick priority values which will be encountered in the metafile. It need not represent the exact priorities in the metafile. The lowest display priority value is zero.

References:

4.3
4.12.4.4
4.12.4.5

Page 56

Add the following note at the end of 5.4.1 (SCALING MODE):

NOTE - If both device viewport and scaling mode appear in the same metafile, the last specified is used. If neither appear, the default values for device viewport take precedence.

Page 58

Add the following sub-clauses after 5.4.7:

5.4.8 DEVICE VIEWPORT

Parameters:

first corner (VP)
second corner (VP)

Description:

The two parameters define the opposite corners of a rectangular viewport on the device's display surface. These parameters are specified by the unit system selected by DEVICE VIEWPORT SPECIFICATION MODE.

The effective viewport is that area of the display surface onto which the VDC extent rectangle is mapped. If the current DEVICE VIEWPORT MAPPING forces isotropic mapping, and the aspect ratio is not equal to that of the device viewport, the effective viewport will be smaller than the specified viewport on one or the other axis (but not both).

If the current DEVICE VIEWPORT MAPPING does not force isotropic mapping, the effective viewport will be the same as the specified viewport. If the Device Viewport exceeds the available display surface, the Device Viewport is still used to determine the VDC-to-Device mapping.

Mirroring or 180^{o} rotation of the image may be achieved by specifying the corners in some way other than that the first is below and to the left of the second.

NOTE - If both device viewport and scaling mode appear in the same metafile, the last specified is used. If neither appear, the default values for device viewport take precedence.

Reference:

4.4.7

5.4.9 DEVICE VIEWPORT SPECIFICATION MODE

Parameters:

VC specifier (one of: fraction of display surface,
millimetres with scalefactor,
physical device coordinates)(E)

metric scale factor (R)

Description:

This element determines how subsequent elements using the data type VC (viewport coordinate) or VP (viewport point) will be defined.

These parameters may be specified in one of three modes: fraction of display surface; millimetres with scale factor; or physical device coordinates.

When the VC specifier is 'fraction of display surface ', the value (0.0, 0.0) corresponds to the lower left corner and the value (1.0, 1.0) corresponds to the upper right corner of the default device viewport. (The default device viewport is the largest unrotated rectangular area visible on the display surface). Numbers outside the range [0.0 to 1.0] may be specified (see 5.4.8). When the VC specifier is 'fraction of display surface' the metric scale factor is ignored.

When the VC specifier is 'millimetres with scalefactor', the metric scale factor parameter represents the distance (in millimetres) on the display surface corresponding to one unit in VC space. One unit in VC space represents one millimetre multiplied by the metric scale factor. The value (0,0) corresponds to the lower left corner and the values increase positively to the right and upwards.

When the VC specifier is 'physical device coordinates', the native units and handedness of the physical device are used. The metric scale factor is ignored.

Metric scaling with a scale factor provides a device-independent means of generating output at a known size. In metric mode, a scale factor of 1.0 indicates that the VC are in units of millimetres; a scale factor of 0.0254 would imply a VC of one thousand per inch.

Reference:
4.4.7

5.4.10 DEVICE VIEWPORT MAPPING

Parameters:

isotropy flag (one of: not forced, forced)(E)
horizontal alignment flag (one of: left, centre, right)(E)
vertical alignment flag (one of: bottom, centre, top)(E).

Description:
This element determines how the coordinate mapping is derived from the VDC EXTENT and the specified DEVICE VIEWPORT. The remaining parameters are significant only if isotropy is forced by the first parameter. If so, the effective viewport is generally smaller than the specified viewport, and these parameters determine how it will be positioned within the specified viewport. 'Left' and 'bottom' are interpreted as being towards the "first corner" of the specified DEVICE VIEWPORT, regardless of any mirroring or rotation of the viewport on the physical device.

Reference:
4.4.7

5.4.11 LINE REPRESENTATION

Parameters:

line bundle index (IX)
line type (IX)
line width specifier
    if line width specification mode is 'absolute',
    absolute line width (VDC)

    if line width specification mode is 'scaled',
    line width scale factor (R)

line colour specifier
    if the colour selection mode is 'indexed',
    line colour index (CI)

    if the colour selection mode is 'direct',
    line colour value (CD)

Description:

In the line bundle table, the given line bundle index is associated with the specific parameters.

Line type is specified and behaves as indicated in the LINE TYPE attribute element.

Line width is defined in the current LINE WIDTH SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes in the specification mode.

Line colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs; see the ASPECT SOURCE FLAG element.

Reference:

4.4.8

5.4.12. MARKER REPRESENTATION

Parameters:

marker bundle index (IX)
marker type (IX)
marker size specifier
  if marker size specification mode is 'absolute',
  absolute marker size (VDC)

  if marker size specification mode is 'scaled',
  marker size scale factor (R)

marker colour specifier
  if the colour selection mode is 'indexed',
  marker colour index (CI)

  if the colour selection mode is 'direct',
  marker colour value (CD)

Description:

In the marker bundle table, the given marker bundle index is associated with the specified parameters.

Marker type is specified and behaves as indicated in the MARKER TYPE attribute element.

Marker size is defined in the current MARKER SIZE SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes in the specification mode.

Marker colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs; see the ASPECT SOURCE FLAG element.

Reference:

4.4.8

5.4.13 TEXT REPRESENTATION

Parameters:

text bundle index (IX)
font index (IX)
text precision (one of: string, character, stroke) (E)
character spacing (R)
character expansion factor (R)
text colour specifier
  if the colour selection mode is 'indexed',
  text colour index (CI)

  if the colour selection mode is 'direct',
  text colour value (CD)

Description:

In the text bundle table, the given text bundle index is associated with the specified parameters.

Font index is specified and behaves as indicated in the TEXT FONT INDEX attribute element.

Text precision is specified and behaves as indicated in the TEXT PRECISION attribute element.

Character spacing is specified and behaves as indicated in the CHARACTER SPACING attribute element.

Character expansion factor is specified and behaves as indicated in the CHARACTER EXPANSION FACTOR attribute element.

Text colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs; see the ASPECT SOURCE FLAG element.

Reference:

4.4.8

5.4.14 FILL REPRESENTATION

Parameters:

fill bundle index (IX)
interior style (one of: hollow, solid, pattern, hatch, empty)(E)
fill colour specifier
  if the colour selection mode is 'indexed',
  fill colour index (CI)

  if the colour selection mode is 'direct',
  fill colour value (CD)

hatch index (IX)
pattern index (IX)

Description:

In the fill bundle table, the given fill bundle index is associated with the specified parameters.

Interior style is specified and behaves as indicated in the INTERIOR STYLE attribute element.

Fill colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Hatch index is specified and behaves as indicated in the HATCH INDEX attribute element.

Pattern index is specified and behaves as indicated in the PATTERN INDEX attribute element.

Which aspects are used depends on the corresponding ASFs; see the ASPECT SOURCE FLAG element.

Reference:
4.4.8

5.4.15 EDGE REPRESENTATION

Parameters:

edge bundle index (IX)
edge type (IX)
edge width specifier
if edge width specification mode is 'absolute',
absolute edge width (VDC)

if edge width specification mode is 'scaled',
edge width scale factor (R)

edge colour specifier
if the colour selection mode is 'indexed',
edge colour index (CI)

if the colour selection mode is 'direct',
edge colour value (CD)

Description:
In the edge bundle table, the given edge bundle index is associated with the specified parameters.

Edge type is specified and behaves as indicated in the EDGE TYPE attribute element.

Edge width is defined in the current EDGE WIDTH SPECIFICATION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes in the specification mode.

Edge colour is defined in the current COLOUR SELECTION MODE and is stored in the bundle table along with that mode. Thus, the definition is immune to subsequent changes to the selection mode.

Which aspects are used depends on the corresponding ASFs; see the ASPECT SOURCE FLAG element.

Reference:
4.4.8

Page 61

Add the following after 5.5.6:

5.5.7 LINE CLIPPING MODE

Parameters:

mode (one of: locus, shape, locus then shape) (E)

Description:
The Line Clipping Mode is set to the value specified.

References:
4.5.2
4.6.1.3

5.5.8 MARKER CLIPPING MODE

Parameters:

mode (one of: locus, shape, locus then shape) (E)

Description:
The Marker Clipping Mode is set to the value specified.

References:
4.5.2
4.6.2.3

5.5.9 EDGE CLIPPING MODE

Parameters:

mode (one of: locus, shape, locus then shape) (E)

Description:
The Edge Clipping Mode is set to the value specified.

References:
4.5.2
4.6.4.5

5.5.10 NEW REGION

Parameters:

none

Description:
This element is used for control of subregion construction within closed figures.

If the current region has not yet been closed by a preceding NEW REGION element and if the last point of the last line element is not coincident with the current closure point, then the current subregion is closed by a line segment connecting the last point of the preceding line element to the current closure point. This line becomes a part of the implicit boundary specification. If the NEW REGION was preceded by a CONNECTING EDGE element, which was itself preceded by a line primitive, then this line also becomes part of the edge specification. If the region which has been previously closed is empty, or if the last point of the last line element is coincident with the current closure point, or if the last element was a filled-area primitive then no line segment is generated by this element.

The first point of the next line element following a NEW REGION element becomes the new closure point, starting a new subregion.

Reference:
4.6.8

5.5.11 SAVE PRIMITIVE CONTEXT

Parameters:

context name (N)

Description:
This element allows for the grouping and identification of the set of current values of the attribute and control elements listed in the list below as a single named entity.

Groups of elements may be saved in a picture or segment (local or global) using the context name.

The attribute and control elements which may be saved by SAVE PRIMITIVE CONTEXT and restored by RESTORE PRIMITIVE CONTEXT are:

LINE BUNDLE INDEX
LINE TYPE
LINE WIDTH (Note 1)
LINE COLOUR (Note 1)
LINE CLIPPING MODE

MARKER BUNDLE INDEX
MARKER TYPE
MARKER SIZE (Note 1)
MARKER COLOUR (Note 1)
MARKER CLIPPING MODE

TEXT BUNDLE INDEX
TEXT FONT INDEX
TEXT PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR (Note 1)
CHARACTER HEIGHT
CHARACTER ORIENTATION
TEXT PATH
TEXT ALIGNMENT
CHARACTER SET INDEX
ALTERNATE CHARACTER SET INDEX

FILL BUNDLE INDEX
INTERIOR STYLE
FILL COLOUR (Note 1)
HATCH INDEX
PATTERN INDEX
EDGE BUNDLE INDEX
EDGE TYPE
EDGE WIDTH (Note 1)
EDGE COLOUR (Note 1)
EDGE VISIBILITY
EDGE CLIPPING MODE
FILL REFERENCE POINT (Note 2)
PATTERN SIZE

PICK IDENTIFIER

CLIP INDICATOR
CLIP RECTANGLE (Note 2)
AUXILIARY COLOUR (Note 1)
TRANSPARENCY

ASPECT SOURCE FLAGS

NOTES

1 The corresponding specification mode or selection mode in which this value was last set is also recorded., this will not cause an implicit change of mode on interpretation of RESTORE PRIMITIVE CONTEXT (see 4.5.3).

2 The VDC TYPE in effect when these values are saved is also recorded.

Reference:

4.5.3

5.5.12 RESTORE PRIMITIVE CONTEXT

Parameters:

context name (N)

Description:

The attribute and control set recorded in the metafile in the saved context name set by the SAVE PRIMITIVE CONTEXT element is recalled on interpretation.

Reference:

4.5.3

Page 63

Add the following text to the end of the second paragraph of 5.6.3

These instructions for the actual displayed portions of a marker apply only to MARKER CLIPPING MODE 'locus'.

Page 65

Sub-clause 5.6.6: In the NOTE replace the words "into the PICTURE OPEN state" by:

"back to the state that pertained when the text element initiating the string occurred."

Page 78

Add the following after 5.6.19:

5.6.20 CIRCULAR ARC CENTRE REVERSED

Parameters:

centrepoint (P)
DX_start, DY_start, DX_end, DY_end (4VDC)
radius (VDC)

Description:

A circular arc is drawn which is defined as follows:

DX_start and DY_start define a start vector, and DX_end and DY_end define an end vector. The tails of these vectors are placed on the centrepoint. A start ray and end ray are derived from the start and end vectors. The start and end rays are semi-infinite lines from the centrepoint in the directions of the start and end vectors respectively.

The specified radius and centrepoint define a circle. The arc is drawn in the negative angular direction (as defined by VDC EXTENT) from the intersection of the circle and the start ray (as obtained by measuring a distance 'radius' along the start ray from the centrepoint) to the intersection of the circle and the end ray.

The arc is displayed with current line element attributes.

Valid values of the vector components are those which produce vectors of non-zero length.

Valid values of 'radius' are non-negative VDC.

If the start ray and end ray are coincident, it is ambiguous whether the defined arc subtends 0° or 360° of central angle (see the specifications for the CIRCULAR ARC CENTRE in annex D).

Reference:

4.6

5.6.21 CONNECTING EDGE

Parameters:

none

Description:

During the construction of a closed figure a line segment connecting the last point of the preceding line element and the next point is added to the boundary and edge definitions. The next point may be either:

1) the first point of the next line element, or

2) the current closure point (in cases where CONNECTING EDGE is followed by either NEW REGION or END FIGURE).

The appearance of the connecting edge is fully determined by the edge attributes including EDGE VISIBILITY.

Reference:
4.6.8

Page 98

Add the following after 5.7.35:

5.7.36 PICK IDENTIFIER

Parameters:

pick identifier (N)

Description:
The pick identifier value is associated with all of the graphical primitive elements of a segment until the next PICK IDENTIFIER element. Usage of the PICK IDENTIFIER on interpretation is dependent upon the application.

Reference:
4.7.9

Page 100

Add the following after 5.9:

5.10 Segment elements

5.10.1 Segment control elements

5.10.1.1 COPY SEGMENT

Parameters:

segment identifier (N)
copy transformation matrix:
  scaling and rotation portion (2 x 2) (R)
  translation portion (2 x 1) (VDC)
segment transformation application (one of: no, yes) (E)

Description:
The segment which is indicated by the segment identifier is referenced at this point in the metafile for copying into the picture, or into a segment when referenced from a segment, on interpretation. The identified segment is referred to as the copied segment. With the possible exception of the segment transformation associated with the copied segment the segment attributes of the copied segment are ignored. The segment attributes of a segment in which the COPY SEGMENT may occur are unchanged by this element.

The copy transformation is applied to all graphic objects of the copied segment before they are copied into the picture or into the segment. The copy transformation is also applied to clipping rectangles under some circumstances.

The INHERITANCE FILTER element allows for control of the control and attribute values which are used when copying segments. This filter controls whether values of individual attribute and control elements are reapplied to the graphic objects. The effects of INHERITANCE FILTER are described in 4.12.5. The way in which clipping is applied to primitives within a copied segment is controlled by CLIP INHERITANCE (see 4.12.5).

The 'segment transformation application' parameter controls whether or not the segment transformation associated with the copied segment will be applied as an effect of the copy process. In no case is the segment transformation applied to a clip rectangle associated with a copied graphic object. In case the 'segment transformation application' is 'yes', the segment transformation is applied prior to the copy transformation.

References:

4.12.1
4.12.5

5.10.1.2 INHERITANCE FILTER

Parameters:

filter selection list (list of elements or groups from:

LINE BUNDLE INDEX
LINE TYPE
LINE WIDTH
LINE COLOUR
LINE CLIPPING MODE
MARKER BUNDLE INDEX
MARKER TYPE
MARKER SIZE
MARKER COLOUR
MARKER CLIPPING MODE
TEXT BUNDLE INDEX
TEXT FONT INDEX
TEXT PRECISION
CHARACTER EXPANSION FACTOR
CHARACTER SPACING
TEXT COLOUR
CHARACTER HEIGHT
CHARACTER ORIENTATION
TEXT PATH
TEXT ALIGNMENT
FILL BUNDLE INDEX
INTERIOR STYLE
FILL COLOUR
HATCH INDEX
PATTERN INDEX
EDGE BUNDLE INDEX
EDGE TYPE
EDGE WIDTH
EDGE COLOUR
EDGE VISIBILITY
EDGE CLIPPING MODE
FILL REFERENCE POINT
PATTERN SIZE
AUXILIARY COLOUR
TRANSPARENCY
LINE ATTRIBUTES
MARKER ATTRIBUTES
TEXT REPRESENTATION AND PLACEMENT ATTRIBUTES
TEXT PLACEMENT AND ORIENTATION ATTRIBUTES
FILL ATTRIBUTES
EDGE ATTRIBUTES
PATTERN ATTRIBUTES
OUTPUT CONTROL
PICK IDENTIFIER
ALL ATTRIBUTES AND CONTROL
ALL
LINE TYPE ASF
LINE WIDTH ASF
LINE COLOUR ASF
MARKER TYPE ASF
MARKER SIZE ASF

MARKER COLOUR ASF
TEXT FONT INDEX ASF
TEXT PRECISION ASF
CHARACTER EXPANSION FACTOR ASF
CHARACTER SPACING ASF
TEXT COLOUR ASF
INTERIOR STYLE ASF
FILL COLOUR ASF
HATCH INDEX ASF
PATTERN INDEX ASF
EDGE TYPE ASF
EDGE WIDTH ASF
EDGE COLOUR ASF
LINE ASFS
MARKER ASFS
TEXT ASFS
FILL ASFS
EDGE ASFS
ALL ASFS) (nE)

selection setting (one of: state list, segment) (E)

Description:

The setting of the inheritance filter is modified for those attributes in the filter selection list. Attributes may be inherited from the modal state lists or from the copied segment depending on the selection setting.

Reference:

4.12.5

5.10.1.3 CLIP INHERITANCE

Parameters:

clip inheritance (one of: state list, intersection) (E)

Description:

The behaviour of clipping as applied to graphic objects in copied segments is defined. Simple clipping against the current rectangle in the modal state list is selected by the value 'state list'. The value 'intersection' not only selects the clip rectangle to come from the segment but also enables an "object clipping" feature. The transformation of clip rectangles and accumulation or composition of multiple transformed rectangles is enabled, depending upon the settings of CLIP INDICATOR (see 4.12.5).

References:

4.12.5

5.10.2 Segment Attribute Elements

Segment Attribute Elements, if used, shall all appear immediately after BEGIN SEGMENT, before the first element of another type. The segment identifier shall refer to the segment in which the elements are contained.

5.10.2.1 SEGMENT TRANSFORMATION

Parameters:

segment identifier (N)
transformation matrix:
scaling and rotation portion (2 x 2) (R)
translation portion (2 x 1) (VDC)

Description:
The segment transformation matrix for the identified segment is set to the specified parameter.

Reference:
4.12.4.2

5.10.2.2 SEGMENT HIGHLIGHTING

Parameters:

segment identifier (N)
highlighting (one of: normal, highlighted) (E)

Description:
The segment highlighting for the identified segment is set to the specified value. When the highlighting attribute is set to 'highlighted', the visual appearance of the segment is interpretation dependent. When the highlighting attribute is set to 'normal', the segment is displayed according to the segment and primitive attributes.

Reference:
4.12.4.3

5.10.2.3 SEGMENT DISPLAY PRIORITY

Parameters:

segment identifier (N)
segment display priority (I)

Description:
The segment display priority for the identified segment is set to the specified value.

Segments with higher segment display priority appear to be in front of segments with lower segment display priorities when displayed following interpretation. When the segment display priorities of two overlapping segments are the same, the order in which they appear is interpretation dependent.

Reference:
4.12.4.4

5.10.2.4 SEGMENT PICK PRIORITY

Parameters:

segment identifier (N)
segment pick priority (I)

Description:
The segment pick priority for the identified segment is set to the specified value. The pick priority does not affect the display of segments.

Reference:
4.12.4.5

Page 103

Clause 6: Add the following at the end:

NAME PRECISION	encoding dependent
MAXIMUM VDC EXTENT	default VDC EXTENT
SEGMENT PRIORITY EXTENT	0...255
DEVICE VIEWPORT	0.,1.,0.,1.
DEVICE VIEWPORT SPECIFICATION MODE	fraction of display surface
DEVICE VIEWPORT MAPPING	forced,left,bottom
LINE REPRESENTATION	interpreter dependent
MARKER REPRESENTATION	interpreter dependent
TEXT REPRESENTATION	interpreter dependent
FILL REPRESENTATION	interpreter dependent
EDGE REPRESENTATION	interpreter dependent
LINE CLIPPING MODE	locus
MARKER CLIPPING MODE	locus
EDGE CLIPPING MODE	locus
PICK IDENTIFIER	0
INHERITANCE FILTER	segment
CLIP INHERITANCE	state list
SEGMENT TRANSFORMATION	1.,0. 0.,1. 0.,0.
SEGMENT HIGHLIGHTING	normal
SEGMENT DISPLAY PRIORITY	0
SEGMENT PICK PRIORITY	0

Page 104

Add the following after 7.4:

7.5 Conformance for Version 2 metafiles

This conformance section defines conformance for metafiles which are 'version 2'. A Computer Graphics Metafile (CGM) is said to conform to the standard if it implements precisely all the elements required for a version 2 metafile as defined in this standard. When determining conformance of a version 2 CGM, the formal grammar shall take precedence.

Page 123

Add the following to the end of sub-clause D.1:

Dynamic effects are avoided by limiting the position of elements with potentially dynamic effects. Thus, bundle table definitions may appear only in the picture descriptor. In a metafile the effects of COLOUR TABLE and PATTERN TABLE are unspecified when they occur in a location with potentially dynamic implications. In metafiles which have a version number which is greater than 1 these elements may appear in the Picture Descriptor. Use of these elements in the picture body is discouraged in order to improve the portability and predictability of CGM exchange.

Page 125

Add the following after D.3.2.2:

D.3.2.3 Order of metafile descriptor elements

It is recommended that the mandatory elements in the Metafile Descriptor are written first in the descriptor and in the following order:

METAFILE VERSION
METAFILE ELEMENT LIST
METAFILE DESCRIPTION

Page 125

Add the following after D.3.2:

D.3.3 Unsatisfied references

Elements referring to a non-existing element are ignored.

Examples:

COPY SEGMENT
If the segment identifier refers to a non-existing segment.

RESTORE PRIMITIVE CONTEXT
If an attribute and control set of that context name does not exist.

Page 127

Sub-clause D.4.3: replace the sentence with the following:

DEVICE VIEWPORT, DEVICE VIEWPORT SPECIFICATION MODE, DEVICE VIEWPORT MAPPING
In the case where the VC specifier in DEVICE VIEWPORT SPECIFICATION MODE is set to either 'millimetres with scale factor' or 'physical device coordinates' not all interpreters may be able to interpret the DEVICE VIEWPORT element as specified, and the interpretation becomes implementation dependent. Since the CGM does not specify the behaviour of an interpreter, an application may wish to control the VDC-to-Device mapping by mechanisms external to the CGM picture description, for example, when including CGM pictures in documents.

Page 127

Add the following text at the end of D.4.4:

CLIPPING MODES
If interpreters cannot implement the 'locus' clipping mode for LINE CLIPPING MODE, MARKER CLIPPING MODE or EDGE CLIPPING MODE, 'locus then shape' should be used as a fallback.

Page 127

Add the following text at the end of the APPEND TEXT recommendations:

Changing the TEXT PRECISION in Text Open (partial text) state is likely to lead to unpredictable results. Generators are discouraged from doing this. Interpreters that can otherwise implement text attribute changes in partial text state should ignore this element in that state as a fallback.

Page 128

Sub-clause D.4.5: Add the following text between CIRCULAR ARC CENTRE CLOSE and Elliptical elements:

CIRCULAR ARC CENTRE REVERSED
If the start ray and end ray coincide, it is recommended that the interpreter draw the full circle.

Page 132

Add the following after D.4.8:

D.4.9 Segment elements

The restriction that segment attributes be set only immediately after the BEGIN SEGMENT element and before any other element avoids any dynamic effects.

SEGMENT DISPLAY PRIORITY
If the output device cannot adjust segment display priority on interpretation, segments should be displayed in order of occurrence.

COPY SEGMENT with CLIP INHERITANCE 'intersection'
If the interpreters cannot handle clip rectangles transformed by a copy transformation with non-zero off-diagonal elements (resulting in a parallelogram) the suggested fallback is to clip to an effective clip rectangle which is the smallest axis-aligned rectangle that contains the transformed clip rectangle. Similarly, in the case where multiple parallelograms might be composed (by intersection) to form a general convex polygon, interpreters should intersect the circumscribing rectangles to derive an effective clip rectangle.

Page 133

Sub-clause D5. Change the words in the first sentence from "..the capabilities shown in table 5" to:

"..the capabilities listed in tables 5a and 5b, appropriate to the version of the metafile they are supporting"

Page 133

Sub-clause D5. Change the title for Table 5 to:

Table 5a - Suggested minimum capabilities for version 1 metafiles.

Page 133

Sub-clause D.5 Add the following table after Table 5a:

Table 5b - Suggested additional minimum capabilities for version 2 metafiles.

Capability	Minimum Suggested Interpreter Support
DEVICE VIEWPORT SPECIFICATION MODE	fraction of display surface
DEVICE VIEWPORT MAPPING	not forced, forced left, centre, right bottom, centre, top
LINE REPRESENTATION	5 entries
MARKER REPRESENTATION	5 entries
TEXT REPRESENTATION	2 entries
FILL REPRESENTATION	5 entries
EDGE REPRESENTATION	5 entries
LINE CLIPPING MODE	locus, shape, locus then shape
MARKER CLIPPING MODE	locus, shape, locus then shape
EDGE CLIPPING MODE	locus, shape, locus then shape
SAVE/RESTORE PRIMITIVE CONTEXT	5 simultaneously saved control and attribute element sets
closed figure	an arbitrary mix containing at least one of the eligible graphical primitives, with POLYGON (SET) supporting at least 128 vertices
segments	64 simultaneously existing segments

Annexes F,G and H are new Annexes and are to be inserted after page 144

Annex F

Formal Grammar of the functional specification of version 2 metafiles

(normative)

F.1 Introduction

This grammar is a formal definition of a standard CGM extended syntax for version 2 metafiles. The encoding-independent and the encoding-dependent productions are separated, and there are subsections showing the syntax of each of the standardized encoding schemes. Details on the encoding of terminal symbols can be found in parts of this Standard that deal with the particular encoding schemes.

F.2 Notation used

```
<symbol>                      - nonterminal
<SYMBOL>                      - terminal
<symbol>*                     - 0 or more occurrences
<symbol>+                     - 1 or more occurrences
<symbol>o                     - optional (0 or 1 occurrences)
<symbol>(n)                   - exactly n occurrences, n=2,3,...
<symbol-1> ::= <symbol-2>     - symbol-1 has the syntax of symbol-2
<symbol-1> | <symbol-2>       - symbol-1 or alternatively symbol-2
<symbol: meaning>             - symbol with the stated meaning
{comment}                     - explanation of a symbol or a production
```

F.3 Detailed grammar

F.3.1 Metafile structure

```
<metafile>                ::= <BEGIN METAFILE>
                                  <metafile identifier>
                              <metafile descriptor>
                              <metafile contents>*
                              <END METAFILE>

<metafile identifier>     ::= <string>

<metafile contents>       ::= <extra element>*
                              <picture>
                              <extra element>*

<extra element>           ::= <external element>
                            | <escape element>

<picture>                 ::= <BEGIN PICTURE>
                                  <picture identifier>
                              <picture descriptor element>*
                              <BEGIN PICTURE BODY>
                              <picture content>*
                              <END PICTURE>

<picture identifier>      ::= <string>

<picture content>         ::= <picture element>
                            | &lt;segment&gt;
```

<picture element>	::=	<control element>
	|	<graphical element>
	|	<closed figure>
	|	<primitive attribute element>
	|	<pattern table element>
	|	<colour table element>
	|	<specification element>
	|	<segment control element>
	|	<extra element>
<segment>	::=	<BEGIN SEGMENT>
		<segment identifier>
		<segment attribute element>*
		<eligible picture element>*
		<END SEGMENT>
<segment identifier>	::=	<name>
<eligible picture element>	::=	<control element>
	|	<graphical element>
	|	<closed figure>
	|	<primitive attribute element>
	|	<specification element>
	|	<segment control element>
	|	<extra element>

F.3.2 Metafile descriptor elements

<metafile descriptor>	::=	<<optional descriptor element>*
		<version>
		<optional descriptor element>*
		<element list>
		<optional descriptor element>*>
	|	<<optional descriptor element>*
		<element list>
		<optional descriptor element>*
		<version>
		<optional descriptor element>*>
<version>	::=	<METAFILE VERSION>
		<integer>
<element list>	::=	<METAFILE ELEMENT LIST>
		<element name>*
	|	<element name shorthand enumerated>*
<element name shorthand enumerated>	::=	<DRAWING SET>
	|	<DRAWING PLUS CONTROL SET>
	|	<VERSION 2 SET>
	|	<EXTENDED PRIMITIVES SET>
	|	<VERSION 2 GKSM SET>
<optional descriptor element>	::=	<description>
	|	<VDC TYPE>
		<vdc type enumerated>
	|	<MAXIMUM COLOUR INDEX>
		<colour index>
	|	<COLOUR VALUE EXTENT>

```
                                      <red green blue>(2)
                                 | <METAFILE DEFAULTS REPLACEMENT>
                                      <element default>+
                                 | <FONT LIST>
                                      <font name>+
                                 | <CHARACTER SET LIST>
                                      <character set definition>+
                                 | <CHARACTER CODING ANNOUNCER>
                                      <coding technique enumerated>
                                 | <scalar precision>
                                 | <MAXIMUM VDC EXTENT>
                                      <point> (2)
                                 | &lt;SEGMENT PRIORITY EXTENT&gt;
                                      <minimum extent>
                                      <maximum extent>
                                 | &lt;segment&gt;
                                 | <extra element>

<description>                  ::= <METAFILE DESCRIPTION>
                                      <string>

<vdc type enumerated>          ::= <INTEGER>
                                 | <REAL>

<element default>              ::= <control element>
                                 | <picture descriptor element>
                                 | <primitive attribute element>
                                 | <extra element>

<font name>                    ::= <string>

<character set definition>     ::= <char set enumerated>
                                   <designation sequence>

<index>                        ::= <standard index value>
                                 | <private index value>

<standard index value>         ::= <positive integer>
<non-negative integer>         ::= <integer>    {greater or equal to 0}
<positive integer>             ::= <integer>    {greater than 0}
<private index value>          ::= <negative integer>
<negative integer>             ::= <integer>    {less than 0}
<positive index>               ::= <positive integer>

<character set enumerated>     ::= <94 CHAR>
                                 | <96 CHAR>
                                 | <MULTI-BYTE 94 CHAR>
                                 | <MULTI-BYTE 96 CHAR>
                                 | <COMPLETE CODE>

<coding technique enumerated>  ::= <BASIC 7-BIT>
                                 | <BASIC 8-BIT>
                                 | <EXTENDED 7-BIT>
                                 | <EXTENDED 8-BIT>

<designation sequence>         ::= <string>

<scalar precision>             ::= <INTEGER PRECISION>
                                      <integer precision value>
                                 | <REAL PRECISION>
                                      <real precision value>
```

```
                          | <INDEX PRECISION>
                                <index precision value>
                          | <COLOUR PRECISION>
                                <colour precision value>
                          | <COLOUR INDEX PRECISION>
                                <colour index precision value>
                          | <NAME PRECISION>
                                <name precision value>
                          {these elements have encoding}
                          {dependent parameters}

<point>                   ::= <vdc value> (2)

<minimum extent>          ::= <integer>

<maximum extent>          ::= <integer>
```

F.3.3 Picture descriptor elements

```
<picture descriptor element>  ::= <SCALING MODE>
                                      <scaling specification mode enumerated>
                                      <metric scale factor>
                                | <VDC EXTENT>
                                      <point> (2)
                                | <DEVICE VIEWPORT>
                                      <viewport point>(2)
                                | <DEVICE VIEWPORT SPECIFICATION MODE
                                      <VC specifier enumerated>
                                      <metric scale factor>
                                | <DEVICE VIEWPORT MAPPING>
                                      <isotropy flag enumerated>
                                      <horizontal alignment flag enumerated>
                                      <vertical alignment flag enumerated>
                                | <BACKGROUND COLOUR>
                                      <red green blue>
                                | <specification element>
                                | <representation element>
                                | <pattern table element>
                                | <colour table element>
                                | <extra element>

<specification element>       ::= <COLOUR SELECTION MODE>
                                      <colour selection mode enumerated>
                                | <LINE WIDTH SPECIFICATION MODE>
                                      <specification mode enumerated>
                                | <MARKER SIZE SPECIFICATION MODE>
                                      <specification mode enumerated>
                                | <EDGE WIDTH SPECIFICATION MODE>
                                      <specification mode enumerated>

<colour selection mode
  enumerated>                 ::= <INDEXED>
                                | <DIRECT>

<scaling specification mode
  enumerated>                 ::= <ABSTRACT>
                                | <METRIC>

<metric scale factor>         ::= <real>

<isotropy flag enumerated>    ::= <NOT FORCED>
```

```
                                  | <FORCED>

<horizontal alignment flag
   enumerated>                  ::= <LEFT>
                                  | <CENTRE>
                                  | <RIGHT>

<vertical alignment flag
   enumerated>                  ::= <BOTTOM>
                                  | <CENTRE>
                                  | <TOP>

<specification mode enumerated> ::= <ABSOLUTE>
                                  | <SCALED>

<viewport point>                ::= <vc value> (2)

<VC specifier enumerated>       ::= <FRACTION OF DISPLAY SURFACE>
                                  | <MILLIMETRES WITH SCALE FACTOR>
                                  | <PHYSICAL DEVICE COORDINATES>

<representation element>        ::= <LINE REPRESENTATION>
                                        <positive index>
                                        <index> {line type}
                                        <size value> {line width}
                                        <colour>
                                  | <MARKER REPRESENTATION>
                                        <positive index>
                                        <index> {marker type}
                                        <size value>
                                        <colour>
                                  | <TEXT REPRESENTATION>
                                        <positive index>
                                        <positive index> {font}
                                        <text precision enumerated>
                                        <real> {character spacing}
                                        <real> {expansion factor}
                                        <colour>
                                  | <FILL REPRESENTATION>
                                        <positive index>
                                        <interior style enumerated>
                                        <colour>
                                        <index> {hatch index}
                                        <positive index> {pattern index}
                                  | <EDGE REPRESENTATION>
                                        <positive index>
                                        <index> {edge type}
                                        <size value> {edge width}
                                        <colour>

<size value>                    ::= <non-negative vdc value>
                                  | <non-negative real>

<non-negative vdc value>        ::= <vdc value> {greater or equal to 0}

<non-negative real>             ::= <real> {greater or equal to 0}

<colour>                        ::= <colour index>
                                  | <red green blue>
```

```
<text precision enumerated>        ::= <STRING>
                                     | <CHARACTER>
                                     | <STROKE>

<interior style enumerated>        ::= <HOLLOW>
                                     | <SOLID>
                                     | <PATTERN>
                                     | <HATCH>
                                     | <EMPTY>
```

F.3.4 Control elements

```
<control element>                  ::= <vdc precision>
                                     | <AUXILIARY COLOUR>
                                           <colour>
                                     | <TRANSPARENCY>
                                           <on-off indicator enumerated>
                                     | <CLIP RECTANGLE>
                                           <point>(2)
                                     | <CLIP INDICATOR>
                                           <on-off indicator enumerated>
                                     | <LINE CLIPPING MODE>
                                           <clip mode enumerated>
                                     | <MARKER CLIPPING MODE>
                                           <clip mode enumerated>
                                     | <EDGE CLIPPING MODE>
                                           <clip mode enumerated>
                                     | <SAVE PRIMITIVE CONTEXT>
                                           <context name>
                                     | <RESTORE PRIMITIVE CONTEXT>
                                           <context name>

<on-off indicator enumerated>      ::= <ON>
                                     | <OFF>

<vdc precision>                    ::= <VDC INTEGER PRECISION>
                                           <vdc integer precision value>
                                     | <VDC REAL PRECISION>
                                           <vdc real precision value>
                                       {these elements have encoding}
                                       {dependent parameters}

<clip mode enumerated>             ::= <LOCUS>
                                     | <SHAPE>
                                     | <LOCUS THEN SHAPE>

<context name>                     ::= <name>
```

F.3.5 Graphical elements

```
<graphical element>                ::= <polypoint element>
                                     | <text element>
                                     | <cell element>
                                     | <gdp element>
                                     | <rectangle element>
                                     | <circular element>
                                     | <elliptical element>
                                     | <pointless element>
```

```
<polypoint element>          ::= <POLYLINE>
                                   <point pair>
                                   <point list>
                               | <DISJOINT POLYLINE>
                                   <point pair>
                                   <point pair list>
                               | <POLYMARKER>
                                   <point>
                                   <point list>
                               | <POLYGON>
                                   <point>(3)
                                   <point list>
                               | <POLYGON SET>
                                   <point edge pair>(3)
                                   <point edge pair list>

<point list>                 ::= <point>*

<point pair list>            ::= <point pair>*

<point pair>                 ::= <point>(2)

<point edge pair>            ::= <point><edge out flag>

<point edge pair list>       ::= <point edge pair>*

<edge out flag>              ::= <INVISIBLE>
                               | <VISIBLE>
                               | <CLOSE INVISIBLE>
                               | <CLOSE VISIBLE>

<text element>               ::= <TEXT>
                                   <point>
                                   <text tail>
                               | <restricted text element>

<restricted text element>    ::= <RESTRICTED TEXT>
                                   <extent>
                                   <point>
                                   <text tail>

<extent>                     ::= <vdc value>(2)

<text tail>                  ::= <final character list>
                               | <nonfinal character list>

<final character list>       ::= <FINAL>
                                   <string>

<nonfinal character list>    ::= <NOT FINAL>
                                   <string>
                                   <partial text attribute element>*
                                   <spanned text>

<spanned text>               ::= <APPEND TEXT>
                                   <text tail>

<cell element>               ::= <CELL ARRAY>
                                   <point>(3)
                                   <integer>(2)
                                   <local colour precision>
```

<colour>(integer1 x integer2)

{this element has an encoding}
{dependent parameter}

<local colour precision> ::= <colour precision value>
| <colour index precision value>
| <default colour precision indicator>

<gdp element> ::= <GDP>
<gdp identifier>
<point list>
<data record>

<gdp identifier> ::= <integer>

<rectangle element> ::= <RECTANGLE>
<point pair>

<circular element> ::= <CIRCLE>
<point>
<radius>
| <CIRCULAR ARC 3 POINT>
<point>(3)
| <CIRCULAR ARC 3 POINT CLOSE>
<point>(3)
<close type>
| <CIRCULAR ARC CENTRE>
<point>
<vdc value>(4)
<radius>
| <CIRCULAR ARC CENTRE CLOSE>
<point>
<vdc value>(4)
<radius>
<close type>
| <CIRCULAR ARC CENTRE REVERSED>
<point>
<vdc value>(4)
<radius>

<radius> ::= <non-negative vdc value>

<close type> ::= <PIE>
| <CHORD>

<elliptical element> ::= <ELLIPSE>
<point>(3)
| <ELLIPTICAL ARC>
<point>(3)
<vdc value>(4)
| <ELLIPTICAL ARC CLOSE>
<point>(3)
<vdc value>(4)
<close type>

<pointless element> ::= <CONNECTING EDGE>

F.3.6 Attribute elements

<primitive attribute element>	::=	<line attribute element>
	|	<marker attribute element>
	|	<text attribute element>
	|	<filled-area attribute element>
	|	<aspect source flags>
	|	<pick identifier>
<line attribute element>	::=	<LINE BUNDLE INDEX> <positive index>
	|	<LINE TYPE> <index>
	|	<LINE WIDTH> <size value>
	|	<LINE COLOUR> <colour>
<marker attribute element>	::=	<MARKER BUNDLE INDEX> <positive index>
	|	<MARKER TYPE> <index>
	|	<MARKER SIZE> <size value>
	|	<MARKER COLOUR> <colour>
<partial text attribute element>	::=	<TEXT FONT INDEX> <positive index>
	|	<TEXT PRECISION> <text precision enumerated>
	|	<CHARACTER EXPANSION FACTOR> <real>
	|	<CHARACTER SPACING> <real>
	|	<TEXT COLOUR> <colour>
	|	<CHARACTER HEIGHT> <non-negative vdc value>
	|	<CHARACTER SET INDEX> <positive index>
	|	<ALTERNATE CHARACTER SET INDEX> <positive index>
	|	<TEXT BUNDLE INDEX> <positive index>
	|	<AUXILIARY COLOUR> <colour>
	|	<TRANSPARENCY> <on-off indicator enumerated>
<text attribute element>	::=	<TEXT BUNDLE INDEX> <positive index>
	|	<TEXT FONT INDEX> <positive index>
	|	<TEXT PRECISION> <text precision enumerated>
	|	<CHARACTER EXPANSION FACTOR> <real>
	|	<CHARACTER SPACING> <real>
	|	<TEXT COLOUR>

```
                                         <colour>
                                   | <CHARACTER HEIGHT>
                                         <non-negative vdc value>
                                   | <CHARACTER ORIENTATION>
                                         <vdc value>(4)
                                   | <TEXT PATH>
                                         <path enumerated>
                                   | <TEXT ALIGNMENT>
                                         <horizontal alignment enumerated>
                                         <vertical alignment enumerated>
                                         <continuous alignment value> (2)
                                   | <CHARACTER SET INDEX>
                                         <positive index>
                                   | <ALTERNATE CHARACTER SET INDEX>
                                         <positive index>

<path enumerated>               ::= <RIGHT>
                                  | <LEFT>
                                  | <UP>
                                  | <DOWN>

<horizontal alignment
   enumerated>                  ::= <NORMAL HORIZONTAL>
                                  | <LEFT>
                                  | <CENTRE>
                                  | <RIGHT>
                                  | <CONTINUOUS HORIZONTAL>

<vertical alignment enumerated> ::= <NORMAL VERTICAL>
                                  | <TOP>
                                  | <CAP>
                                  | <HALF>
                                  | <BASE>
                                  | <BOTTOM>
                                  | <CONTINUOUS VERTICAL>

<continuous alignment value>    ::= <real>

<filled-area attribute element> ::= <FILL BUNDLE INDEX>
                                        <positive index>
                                  | <INTERIOR STYLE>
                                        <interior style enumerated>
                                  | <FILL COLOUR>
                                        <colour>
                                  | <HATCH INDEX>
                                        <index>
                                  | <PATTERN INDEX>
                                        <positive index>
                                  | <EDGE BUNDLE INDEX>
                                        <positive index>
                                  | <EDGE TYPE>
                                        <index>
                                  | <EDGE WIDTH>
                                        <size value>
                                  | <EDGE COLOUR>
                                        <colour>
                                  | <EDGE VISIBILITY>
                                        <on-off indicator enumerated>
                                  | <FILL REFERENCE POINT>
                                        <point>
```

```
                                | <PATTERN SIZE>
                                      <vdc value>(4)

<colour table element>        ::= <COLOUR TABLE>
                                      <starting index>
                                      <red green blue>+

<pattern table element>       ::= <PATTERN TABLE>
                                      <positive index>
                                      <integer>(2)
                                      <local colour precision>
                                      <colour>(integer1 x integer2)
                                  {this element has an encoding}
                                  {dependent parameter}

<starting index>              ::= <colour index>

<aspect source flags>         ::= <ASPECT SOURCE FLAGS>
                                      <asf pair>+

<asf pair>                    ::= <asf type enumerated>
                                  <asf enumerated>

<asf type enumerated>         ::= <LINE TYPE ASF>
                                | <LINE WIDTH ASF>
                                | <LINE COLOUR ASF>
                                | <MARKER TYPE ASF>
                                | <MARKER SIZE ASF>
                                | <MARKER COLOUR ASF>
                                | <TEXT FONT ASF>
                                | <TEXT PRECISION ASF>
                                | <CHARACTER EXPANSION FACTOR ASF>
                                | <CHARACTER SPACING ASF>
                                | <TEXT COLOUR ASF>
                                | <INTERIOR STYLE ASF>
                                | <FILL COLOUR ASF>
                                | <HATCH INDEX ASF>
                                | <PATTERN INDEX ASF>
                                | <EDGE TYPE ASF>
                                | <EDGE WIDTH ASF>
                                | <EDGE COLOUR ASF>

<asf enumerated>              ::= <INDIVIDUAL>
                                | <BUNDLED>

<pick identifier>             ::= <PICK IDENTIFIER>
                                      <name>
```

F.3.7 Closed figure element

```
<closed figure>               ::= <BEGIN FIGURE>
                                  <eligible elements within closed figures>
                                  <END FIGURE>

<eligible elements within
            closed figures>   ::= <VDC REAL PRECISION>
                                | <VDC INTEGER PRECISION>
                                | <AUXILIARY COLOUR>
                                | <TRANSPARENCY>
                                | <NEW REGION>
```

```
                                | <POLYLINE>
                                | <DISJOINT POLYLINE>
                                | <POLYGON>
                                | <POLYGON SET>
                                | <GDP>
                                | <RECTANGLE>
                                | <CIRCLE>
                                | <CIRCULAR ARC 3 POINT>
                                | <CIRCULAR ARC 3 POINT CLOSE>
                                | <CIRCULAR ARC CENTRE>
                                | <CIRCULAR ARC CENTRE CLOSE>
                                | <CIRCULAR ARC CENTRE REVERSED>
                                | <ELLIPSE>
                                | <ELLIPTICAL ARC>
                                | <ELLIPTICAL ARC CLOSE>
                                | <CONNECTING EDGE>
                                | <EDGE BUNDLE INDEX>
                                | <EDGE TYPE>
                                | <EDGE WIDTH>
                                | <EDGE COLOUR>
                                | <EDGE VISIBILITY>
                                | <EDGE TYPE ASF>
                                | <EDGE WIDTH ASF>
                                | <EDGE COLOUR ASF>
                                | <ESCAPE>
                                | <MESSAGE>
                                | <APPLICATION DATA>
```

F.3.8 Escape elements

```
<escape element>              ::= <ESCAPE>
                                      <identifier>
                                      <data record>

<identifier>                  ::= <integer>
```

F.3.9 External elements

```
<external element>            ::= <MESSAGE>
                                      <action flag enumerated>
                                      <string>
                                | <APPLICATION DATA>
                                      <integer>
                                      <data record>

<action flag enumerated>      ::= <YES>
                                | <NO>
```

F.3.10 Segment elements

<segment control element> ::= <COPY SEGMENT>
    <segment identifier>
    <copy transformation matrix>
    <segment transformation application>
| <INHERITANCE FILTER>
    <filter selection list enumerated>*
    <selection setting enumerated>
| <CLIP INHERITANCE>

<clip inheritance enumerated>

<segment attribute element> ::= <SEGMENT TRANSFORMATION>
<segment identifier>
<transformation matrix>
| <SEGMENT HIGHLIGHTING>
<segment identifier>
<highlighting enumerated>
| <SEGMENT DISPLAY PRIORITY>
<segment identifier>
<segment display priority>
| <SEGMENT PICK PRIORITY>
<segment identifier>
<segment pick priority>

<copy transformation matrix> ::= <transformation matrix>

<transformation matrix> ::= <2 x 2 matrix of reals>
<2 x 1 matrix of vdcs>

<segment transformation application> ::= <NO>
| <YES>

<filter selection list enumerated> ::= <attribute and control name enumerated>
| <attribute and control group enumerated>
| <asf name enumerated>
| <asf group enumerated>

<attribute and control name enumerated> ::= <LINE BUNDLE INDEX>
| <LINE TYPE>
| <LINE WIDTH>
| <LINE COLOUR>
| <LINE CLIPPING MODE>
| <MARKER BUNDLE INDEX>
| <MARKER TYPE>
| <MARKER SIZE>
| <MARKER COLOUR>
| <MARKER CLIPPING MODE>
| <TEXT BUNDLE INDEX>
| <TEXT FONT INDEX>
| <TEXT PRECISION>
| <CHARACTER EXPANSION FACTOR>
| <CHARACTER SPACING>
| <TEXT COLOUR>
| <CHARACTER HEIGHT>
| <CHARACTER ORIENTATION>
| <TEXT PATH>
| <TEXT ALIGNMENT>
| <FILL BUNDLE INDEX>
| <INTERIOR STYLE>
| <FILL COLOUR>
| <HATCH INDEX>
| <PATTERN INDEX>
| <EDGE BUNDLE INDEX>
| <EDGE TYPE>
| <EDGE WIDTH>
| <EDGE COLOUR>
| <EDGE VISIBILITY>

	|	<EDGE CLIPPING MODE>
	|	<FILL REFERENCE POINT>
	|	<PATTERN SIZE>
	|	<AUXILIARY COLOUR
	|	<TRANSPARENCY>
<attribute and control group enumerated>	::=	<LINE ATTRIBUTES>
	|	<MARKER ATTRIBUTES>
	|	<TEXT PRESENTATION AND PLACEMENT ATTRIBUTES>
	|	<TEXT PLACEMENT AND ORIENTATION ATTRIBUTES>
	|	<FILL ATTRIBUTES>
	|	<EDGE ATTRIBUTES>
	|	<PATTERN ATTRIBUTES>
	|	<OUTPUT CONTROL>
	|	<PICK IDENTIFIER>
	|	<ALL ATTRIBUTES AND CONTROL>
	|	<ALL>
<selection setting enumerated>	::=	<STATE LIST>
	|	<SEGMENT>
<asf name enumerated>	::=	<LINE TYPE ASF>
	|	<LINE WIDTH ASF>
	|	<LINE COLOUR ASF>
	|	<MARKER TYPE ASF>
	|	<MARKER SIZE ASF>
	|	<MARKER COLOUR ASF>
	|	<TEXT FONT INDEX ASF>
	|	<TEXT PRECISION ASF>
	|	<CHARACTER EXPANSION FACTOR ASF>
	|	<CHARACTER SPACING ASF>
	|	<TEXT COLOUR ASF>
	|	<INTERIOR STYLE ASF>
	|	<FILL COLOUR ASF>
	|	<HATCH INDEX ASF>
	|	<PATTERN INDEX ASF>
	|	<EDGE TYPE ASF>
	|	<EDGE WIDTH ASF>
	|	<EDGE COLOUR ASF>
<asf group enumerated>	::=	<LINE ASFS>
	|	<MARKER ASFS>
	|	<TEXT ASFS>
	|	<FILL ASFS>
	|	<EDGE ASFS>
	|	<ALL ASFS>
<clip inheritance enumerated>	::=	<STATE LIST>
	|	<INTERSECTION>
<highlighting enumerated>	::=	<NORMAL>
	|	<HIGHLIGHTED>
<segment display priority>	::=	<integer>
<segment pick priority>	::=	<integer>

F.4 Terminal symbols

The following are the terminals in this grammar. Their representation is dependent on the encoding scheme used. In annex A of the subsequent parts of this Standard, these encoding-dependent symbols are further described.

<element name>
<integer>
<real>
<vdc value>
<string>
<colour index>
<red green blue>
<integer precision value>
<real precision value>
<index precision value>
<colour precision value>
<colour index precision value>
<name precision value>
<default colour precision indicator>
<vdc integer precision value>
<vdc real precision value>
<data record>
<name>
<vc value>
<2 x 2 matrix of reals>
<2 x 1 matrix of vdcs>

The CGM extended opcodes are encoding dependent. A complete list of them can be found in the productions for <element name enumerated> below.

The enumerated types are:

<INTEGER>
<REAL>
<ON>
<OFF>
<INDEXED>
<DIRECT>
<ABSTRACT>
<METRIC>
<ABSOLUTE>
<SCALED>
<94 CHAR>
<96 CHAR>
<MULTI-BYTE 94 CHAR>
<MULTI-BYTE 96 CHAR>
<COMPLETE CODE>
<BASIC 7-BIT>
<BASIC 8-BIT>
<EXTENDED 7-BIT>
<EXTENDED 8-BIT>
<FRACTION OF DISPLAY SURFACE>
<MILLIMETRES WITH SCALE FACTOR>
<PHYSICAL DEVICE COORDINATES>
<NOT FORCED>
<FORCED>
<LEFT>
<RIGHT>
<CENTRE>
<BOTTOM>

<TOP>
<LOCUS>
<SHAPE>
<LOCUS THEN SHAPE>
<INVISIBLE>
<VISIBLE>
<CLOSE INVISIBLE>
<CLOSE VISIBLE>
<PIE>
<CHORD>
<FINAL>
<NOT FINAL>
<INDIVIDUAL>
<BUNDLED>
<HOLLOW>
<SOLID>
<PATTERN>
<HATCH>
<EMPTY>
<STRING>
<CHARACTER>
<STROKE>
<UP>
<DOWN>
<NORMAL HORIZONTAL>
<CONTINUOUS HORIZONTAL>
<NORMAL VERTICAL>
<CAP>
<HALF>
<BASE>
<CONTINUOUS VERTICAL>
<YES>
<NO>
<LINE TYPE ASF>
<LINE WIDTH ASF>
<LINE COLOUR ASF>
<MARKER TYPE ASF>
<MARKER SIZE ASF>
<MARKER COLOUR ASF>
<TEXT FONT ASF>
<TEXT PRECISION ASF>
<CHARACTER EXPANSION FACTOR ASF>
<CHARACTER SPACING ASF>
<TEXT COLOUR ASF>
<INTERIOR STYLE ASF>
<HATCH INDEX ASF>
<PATTERN INDEX ASF>
<FILL COLOUR ASF>
<EDGE TYPE ASF>
<EDGE WIDTH ASF>
<EDGE COLOUR ASF>
<LINE ATTRIBUTES>
<MARKER ATTRIBUTES>
<TEXT PRESENTATION AND PLACEMENT ATTRIBUTES>
<TEXT PLACEMENT AND ORIENTATION ATTRIBUTES>
<FILL ATTRIBUTES>
<EDGE ATTRIBUTES>
<PATTERN ATTRIBUTES>
<OUTPUT CONTROL>
<ALL ATTRIBUTES AND CONTROL>
<ALL>

<LINE BUNDLE INDEX>
<LINE TYPE>
<LINE WIDTH>
<LINE COLOUR>
<LINE CLIPPING MODE>
<MARKER BUNDLE INDEX>
<MARKER TYPE>
<MARKER SIZE>
<MARKER COLOUR>
<MARKER CLIPPING MODE>
<TEXT BUNDLE INDEX>
<TEXT FONT INDEX>
<TEXT PRECISION>
<CHARACTER EXPANSION FACTOR>
<CHARACTER SPACING>
<TEXT COLOUR>
<CHARACTER HEIGHT>
<CHARACTER ORIENTATION>
<TEXT PATH>
<TEXT ALIGNMENT>
<FILL BUNDLE INDEX>
<INTERIOR STYLE>
<FILL COLOUR>
<HATCH INDEX>
<PATTERN INDEX>
<EDGE BUNDLE INDEX>
<EDGE TYPE>
<EDGE WIDTH>
<EDGE COLOUR>
<EDGE VISIBILITY>
<EDGE CLIPPING MODE>
<FILL REFERENCE POINT>
<PATTERN SIZE>
<AUXILIARY COLOUR>
<TRANSPARENCY>
<STATE LIST>
<INTERSECTION>
<SEGMENT>
<LINE ASFS>
<MARKER ASFS>
<TEXT ASFS>
<FILL ASFS>
<EDGE ASFS>
<ALL ASFS>
<NORMAL>
<HIGHLIGHTED>
<DRAWING SET>
<DRAWING PLUS CONTROL SET>
<VERSION 2 SET>
<EXTENDED PRIMITIVES SET>
<VERSION 2 GKSM SET>

<element name enumerated> ::= <BEGIN METAFILE>
| <END METAFILE>
| <BEGIN PICTURE>
| <BEGIN PICTURE BODY>
| <END PICTURE>
| <BEGIN SEGMENT>
| <END SEGMENT>
| <BEGIN FIGURE>

| <END FIGURE>
| <METAFILE VERSION>
| <METAFILE DESCRIPTION>
| <VDC TYPE>
| <INTEGER PRECISION>
| <REAL PRECISION>
| <INDEX PRECISION>
| <COLOUR PRECISION>
| <COLOUR INDEX PRECISION>
| <NAME PRECISION>
| <MAXIMUM COLOUR INDEX>
| <COLOUR VALUE EXTENT>
| <METAFILE ELEMENT LIST>
| <METAFILE DEFAULTS REPLACEMENT>
| <FONT LIST>
| <CHARACTER SET LIST>
| <CHARACTER CODING ANNOUNCER>
| <MAXIMUM VDC EXTENT>
| <SEGMENT PRIORITY EXTENT>
| <SCALING MODE>
| <COLOUR SELECTION MODE>
| <LINE WIDTH SPECIFICATION MODE>
| <MARKER SIZE SPECIFICATION MODE>
| <EDGE WIDTH SPECIFICATION MODE>
| <VDC EXTENT>
| <BACKGROUND COLOUR>
| <DEVICE VIEWPORT>
| <DEVICE VIEWPORT SPECIFICATION MODE>
| <DEVICE VIEWPORT MAPPING>
| <LINE REPRESENTATION>
| <MARKER REPRESENTATION>
| <TEXT REPRESENTATION>
| <FILL REPRESENTATION>
| <EDGE REPRESENTATION>
| <VDC INTEGER PRECISION>
| <VDC REAL PRECISION>
| <AUXILIARY COLOUR>
| <TRANSPARENCY>
| <CLIP RECTANGLE>
| <CLIP INDICATOR>
| <LINE CLIPPING MODE>
| <MARKER CLIPPING MODE>
| <EDGE CLIPPING MODE>
| <NEW REGION>
| <SAVE PRIMITIVE CONTEXT>
| <RESTORE PRIMITIVE CONTEXT>
| <POLYLINE>
| <DISJOINT POLYLINE>
| <POLYMARKER>
| <TEXT>
| <RESTRICTED TEXT>
| <APPEND TEXT>
| <POLYGON>
| <POLYGON SET>
| <CELL ARRAY>
| <GDP>
| <RECTANGLE>
| <CIRCLE>
| <CIRCULAR ARC 3 POINT>
| <CIRCULAR ARC 3 POINT CLOSE>
| <CIRCULAR ARC CENTRE>

| <CIRCULAR ARC CENTRE CLOSE>
| <CIRCULAR ARC CENTRE REVERSED>
| <ELLIPSE>
| <ELLIPTICAL ARC>
| <ELLIPTICAL ARC CLOSE>
| <CONNECTING EDGE>
| <LINE BUNDLE INDEX>
| <LINE TYPE>
| <LINE WIDTH>
| <LINE COLOUR>
| <MARKER BUNDLE INDEX>
| <MARKER TYPE>
| <MARKER SIZE>
| <MARKER COLOUR>
| <TEXT BUNDLE INDEX>
| <TEXT FONT INDEX>
| <TEXT PRECISION>
| <CHARACTER EXPANSION FACTOR>
| <CHARACTER SPACING>
| <TEXT COLOUR>
| <CHARACTER HEIGHT>
| <CHARACTER ORIENTATION>
| <TEXT PATH>
| <TEXT ALIGNMENT>
| <CHARACTER SET INDEX>
| <ALTERNATE CHARACTER SET INDEX>
| <FILL BUNDLE INDEX>
| <INTERIOR STYLE>
| <FILL COLOUR>
| <HATCH INDEX>
| <PATTERN INDEX>
| <EDGE BUNDLE INDEX>
| <EDGE TYPE>
| <EDGE WIDTH>
| <EDGE COLOUR>
| <EDGE VISIBILITY>
| <FILL REFERENCE POINT>
| <PATTERN TABLE>
| <PATTERN SIZE>
| <COLOUR TABLE>
| <ASPECT SOURCE FLAGS>
| <PICK IDENTIFIER>
| <COPY SEGMENT>
| <INHERITANCE FILTER>
| <CLIP INHERITANCE>
| <SEGMENT TRANSFORMATION>
| <SEGMENT HIGHLIGHTING>
| <SEGMENT DISPLAY PRIORITY>
| <SEGMENT PICK PRIORITY>
| <ESCAPE>
| <MESSAGE>
| <APPLICATION DATA>

Annex G
Formal grammar of the functional specification of version 1 metafiles

(normative)

G.1 Introduction

This grammar is a formal definition of a standard CGM syntax for version 1 metafiles. The encoding-independent and the encoding-dependent productions are separated, and there are subsections showing the syntax of each of the standardized encoding schemes. Details on the encoding of terminal symbols can be found in parts of this Standard that deal with the particular encoding schemes.

G.2 Notation used

<symbol> - nonterminal
<SYMBOL> - terminal
<symbol>* - 0 or more occurrences
<symbol>+ - 1 or more occurrences
<symbol>o - optional (0 or 1 occurrences)
<symbol>(n) - exactly n occurrences, n=2,3,...
<symbol-1> ::= <symbol-2> - symbol-1 has the syntax of symbol-2
<symbol-1> | <symbol-2> - symbol-1 or alternatively symbol-2
<symbol: meaning> - symbol with the stated meaning
{comment} - explanation of a symbol or a production

G.3 Detailed grammar

G.3.1 Metafile structure

<metafile> ::= <BEGIN METAFILE>
<metafile identifier>
<metafile descriptor>
<metafile contents>*
<END METAFILE>

<metafile identifier> ::= <string>

<metafile contents> ::= <extra element>*
<picture>
<extra element>*

<extra element> ::= <external element>
| <escape element>

<picture> ::= <BEGIN PICTURE>
<picture identifier>
<picture descriptor element>*
<BEGIN PICTURE BODY>
<picture element>*
<END PICTURE>

<picture identifier> ::= <string>

<picture element> ::= <control element>

```
                              | <graphical element>
                              | <primitive attribute element>
                              | <pattern table element>
                              | <colour table element>
                              | <extra element>
```

G.3.2 Metafile descriptor elements

```
<metafile descriptor>        ::= <<optional descriptor element>*
                                 <version>
                                 <optional descriptor element>*
                                 <element list>
                                 <optional descriptor element>*>
                               | <<optional descriptor element>*
                                 <element list>
                                 <optional descriptor element>*
                                 <version>
                                 <optional descriptor element>*>

<version>                    ::= <METAFILE VERSION>
                                     <integer>

<element list>               ::= <METAFILE ELEMENT LIST>
                                     <element name>*
                               |     <element name shorthand enumerated>*

<element name shorthand
      enumerated>            ::= <DRAWING SET>
                               | <DRAWING PLUS CONTROL SET>

<optional descriptor element> ::= <description>
                               | <VDC TYPE>
                                     <vdc type enumerated>
                               | <MAXIMUM COLOUR INDEX>
                                     <colour index>
                               | <COLOUR VALUE EXTENT>
                                     <red green blue>(2)
                               | <METAFILE DEFAULTS REPLACEMENT>
                                     <element default>+
                               | <FONT LIST>
                                     <font name>+
                               | <CHARACTER SET LIST>
                                     <character set definition>+
                               | <CHARACTER CODING ANNOUNCER>
                                     <coding technique enumerated>
                               | <scalar precision>
                               | <extra element>

<description>                ::= <METAFILE DESCRIPTION>
                                     <string>

<vdc type enumerated>        ::= <INTEGER>
                               | <REAL>

<element default>            ::= <control element>
                               | <picture descriptor element>
                               | <primitive attribute element>
                               | <extra element>

<font name>                  ::= <string>
```

```
<character set definition>      ::= <character set enumerated>
                                    <designation sequence>

<index>                         ::= <standard index value>
                                  | <private index value>

<standard index value>          ::= <positive integer>
<non-negative integer>          ::= <integer>   {greater or equal to 0}
<positive integer>              ::= <integer>   {greater than 0}
<private index value>           ::= <negative integer>
<negative integer>              ::= <integer>   {less than 0}
<positive index>                ::= <positive integer>

<character set enumerated>      ::= <94 CHAR>
                                  | <96 CHAR>
                                  | <MULTI-BYTE 94 CHAR>
                                  | <MULTI-BYTE 96 CHAR>
                                  | <COMPLETE CODE>

<coding technique enumerated>   ::= <BASIC 7-BIT>
                                  | <BASIC 8-BIT>
                                  | <EXTENDED 7-BIT>
                                  | <EXTENDED 8-BIT>

<designation sequence>          ::= <string>

<scalar precision>              ::= <INTEGER PRECISION>
                                        <integer precision value>
                                  | <REAL PRECISION>
                                        <real precision value>
                                  | <INDEX PRECISION>
                                        <index precision value>
                                  | <COLOUR PRECISION>
                                        <colour precision value>
                                  | <COLOUR INDEX PRECISION>
                                        <colour index precision value>
                                    {these elements have encoding}
                                    {dependent parameters}
```

G.3.3 Picture descriptor elements

```
<picture descriptor element>    ::= <SCALING MODE>
                                        <scaling specification mode enumerated>
                                        <metric scale factor>
                                  | <VDC EXTENT>
                                        <point> (2)
                                  | <BACKGROUND COLOUR>
                                        <red green blue>
                                  | <specification element>
                                  | <extra element>

<specification element>         ::= <COLOUR SELECTION MODE>
                                        <colour selection mode enumerated>
                                  | <LINE WIDTH SPECIFICATION MODE>
                                        <specification mode enumerated>
                                  | <MARKER SIZE SPECIFICATION MODE>
                                        <specification mode enumerated>
                                  | <EDGE WIDTH SPECIFICATION MODE>
                                        <specification mode enumerated>
```

```
<colour selection mode
  enumerated>              ::= <INDEXED>
                             | <DIRECT>

<scaling specification mode
  enumerated>              ::= <ABSTRACT>
                             | <METRIC>

<metric scale factor>      ::= <real>

<specification mode enumerated> ::= <ABSOLUTE>
                             | <SCALED>

<point>                    ::= <vdc value> (2)
```

G.3.4 Control elements

```
<control element>          ::= <vdc precision>
                             | <AUXILIARY COLOUR>
                                   <colour>
                             | <TRANSPARENCY>
                                   <on-off indicator enumerated>
                             | <CLIP RECTANGLE>
                                   <point>(2)
                             | <CLIP INDICATOR>
                                   <on-off indicator enumerated>

<on-off indicator enumerated> ::= <ON>
                             | <OFF>

<colour>                   ::= <colour index>
                             |<red green blue>

<vdc precision>            ::= <VDC INTEGER PRECISION>
                                   <vdc integer precision value>
                             | <VDC REAL PRECISION>
                                   <vdc real precision value>
                               {these elements have encoding}
                               {dependent parameters}
```

G.3.5 Graphical elements

```
<graphical element>        ::= <polypoint element>
                             | <text element>
                             | <cell element>
                             | <gdp element>
                             | <rectangle element>
                             | <circular element>
                             | <elliptical element>

<polypoint element>        ::= <POLYLINE>
                                   <point pair>
                                   <point list>
                             | <DISJOINT POLYLINE>
                                   <point pair>
                                   <point pair list>
                             | <POLYMARKER>
                                   <point>
                                   <point list>
                             | <POLYGON>
```

```
                                <point>(3)
                                <point list>
                            | <POLYGON SET>
                                <point edge pair>(3)
                                <point edge pair list>

<point list>                ::= <point>*

<point pair list>           ::= <point pair>*

<point pair>                ::= <point>(2)

<point edge pair>           ::= <point><edge out flag>

<point edge pair list>      ::= <point edge pair>*

<edge out flag>             ::= <INVISIBLE>
                            | <VISIBLE>
                            | <CLOSE INVISIBLE>
                            | <CLOSE VISIBLE>

<text element>              ::= <TEXT>
                                <point>
                                <text tail>
                            | <restricted text element>

<restricted text element>   ::= <RESTRICTED TEXT>
                                <extent>
                                <point>
                                <text tail>

<extent>                    ::= <vdc value>(2)

<text tail>                 ::= <final character list>
                            | <nonfinal character list>

<final character list>      ::= <FINAL>
                                <string>

<nonfinal character list>   ::= <NOT FINAL>
                                <string>
                                <partial text attribute element>*
                                <spanned text>

<spanned text>              ::= <APPEND TEXT>
                                <text tail>

<cell element>              ::= <CELL ARRAY>
                                <point>(3)
                                <integer>(2)
                                <local colour precision>
                                <colour>(integer1 x integer2)

                            {this element has an encoding}
                            {dependent parameter}

<local colour precision>    ::= <colour precision value>
                            | <colour index precision value>
                            | <default colour precision indicator>

<gdp element>               ::= <GDP>
```

```
                                      <gdp identifier>
                                      <point list>
                                      <data record>

<gdp identifier>             ::=  <integer>

<rectangle element>          ::=  <RECTANGLE>
                                      <point pair>

<circular element>           ::=  <CIRCLE>
                                      <point>
                                      <radius>
                               |  <CIRCULAR ARC 3 POINT>
                                      <point>(3)
                               |  <CIRCULAR ARC 3 POINT CLOSE>
                                      <point>(3)
                                      <close type>
                               |  <CIRCULAR ARC CENTRE>
                                      <point>
                                      <vdc value>(4)
                                      <radius>
                               |  <CIRCULAR ARC CENTRE CLOSE>
                                      <point>
                                      <vdc value>(4)
                                      <radius>
                                      <close type>

<radius>                     ::=  <non-negative vdc value>

<non-negative vdc value>     ::=  <vdc value> {greater than or equal to 0}

<close type>                 ::=  <PIE>
                               |  <CHORD>

<elliptical element>         ::=  <ELLIPSE>
                                      <point>(3)
                               |  <ELLIPTICAL ARC>
                                      <point>(3)
                                      <vdc value>(4)
                               |  <ELLIPTICAL ARC CLOSE>
                                      <point>(3)
                                      <vdc value>(4)
                                      <close type>
```

G.3.6 Attribute elements

```
<primitive attribute element> ::=  <line attribute element>
                               |  <marker attribute element>
                               |  <text attribute element>
                               |  <filled-area attribute element>
                               |  <aspect source flags>

<line attribute element>     ::=  <LINE BUNDLE INDEX>
                                      <positive index>
                               |  <LINE TYPE>
                                      <index>
                               |  <LINE WIDTH>
                                      <size value>
                               |  <LINE COLOUR>
                                      <colour>
```

```
<size value>                     ::= <non-negative vdc value>
                                   | <non-negative real>

<non-negative real>              ::= <real> {greater than or equal to 0}

<marker attribute element>       ::= <MARKER BUNDLE INDEX>
                                         <positive index>
                                   | <MARKER TYPE>
                                         <index>
                                   | <MARKER SIZE>
                                         <size value>
                                   | <MARKER COLOUR>
                                         <colour>

<partial text attribute element> ::= <TEXT FONT INDEX>
                                         <positive index>
                                   | <TEXT PRECISION>
                                         <text precision enumerated>
                                   | <CHARACTER EXPANSION FACTOR>
                                         <real>
                                   | <CHARACTER SPACING>
                                         <real>
                                   | <TEXT COLOUR>
                                         <colour>
                                   | <CHARACTER HEIGHT>
                                         <non-negative vdc value>
                                   | <CHARACTER SET INDEX>
                                         <positive index>
                                   | <ALTERNATE CHARACTER SET INDEX>
                                         <positive index>
                                   | <TEXT BUNDLE INDEX>
                                         <positive index>
                                   | <AUXILIARY COLOUR>
                                         <colour>
                                   | <TRANSPARENCY>
                                         <on-off indicator enumerated>

<text attribute element>         ::= <TEXT BUNDLE INDEX>
                                         <positive index>
                                   | <TEXT FONT INDEX>
                                         <positive index>
                                   | <TEXT PRECISION>
                                         <text precision enumerated>
                                   | <CHARACTER EXPANSION FACTOR>
                                         <real>
                                   | <CHARACTER SPACING>
                                         <real>
                                   | <TEXT COLOUR>
                                         <colour>
                                   | <CHARACTER HEIGHT>
                                         <non-negative vdc value>
                                   | <CHARACTER ORIENTATION>
                                         <vdc value>(4)
                                   | <TEXT PATH>
                                         <path enumerated>
                                   | <TEXT ALIGNMENT>
                                         <horizontal alignment enumerated>
                                         <vertical alignment enumerated>
                                         <continuous alignment value> (2)
                                   | <CHARACTER SET INDEX>
                                         <positive index>
```

```
                                   | <ALTERNATE CHARACTER SET INDEX>
                                          <positive index>

<text precision enumerated>    ::=  <STRING>
                                 |  <CHARACTER>
                                 |  <STROKE>

<path enumerated>              ::=  <RIGHT>
                                 |  <LEFT>
                                 |  <UP>
                                 |  <DOWN>

<horizontal alignment enumerated>  ::=  <NORMAL HORIZONTAL>
                                 |  <LEFT>
                                 |  <CENTRE>
                                 |  <RIGHT>
                                 |  <CONTINUOUS HORIZONTAL>

<vertical alignment enumerated>  ::=  <NORMAL VERTICAL>
                                 |  <TOP>
                                 |  <CAP>
                                 |  <HALF>
                                 |  <BASE>
                                 |  <BOTTOM>
                                 |  <CONTINUOUS VERTICAL>

<continuous alignment value>   ::=  <real>

<filled-area attribute element>  ::=  <FILL BUNDLE INDEX>
                                          <positive index>
                                 |  <INTERIOR STYLE>
                                          <interior style enumerated>
                                 |  <FILL COLOUR>
                                          <colour>
                                 |  <HATCH INDEX>
                                          <index>
                                 |  <PATTERN INDEX>
                                          <positive index>
                                 |  <EDGE BUNDLE INDEX>
                                          <positive index>
                                 |  <EDGE TYPE>
                                          <index>
                                 |  <EDGE WIDTH>
                                          <size value>
                                 |  <EDGE COLOUR>
                                          <colour>
                                 |  <EDGE VISIBILITY>
                                          <on-off indicator enumerated>
                                 |  <FILL REFERENCE POINT>
                                          <point>
                                 |  <PATTERN SIZE>
                                          <vdc value>(4)

<interior style enumerated>    ::=  <HOLLOW>
                                 |  <SOLID>
                                 |  <PATTERN>
                                 |  <HATCH>
                                 |  <EMPTY>

<colour table element>         ::=  <COLOUR TABLE>
```

```
                                        <starting index>
                                        <red green blue>+

<pattern table element>          ::=  <PATTERN TABLE>
                                        <positive index>
                                        <integer>(2)
                                        <local colour precision>
                                        <colour>(integer1 x integer2)
                                      {this element has an encoding}
                                      {dependent parameter}

<starting index>                 ::=  <colour index>

<aspect source flags>            ::=  <ASPECT SOURCE FLAGS>
                                        <asf pair>+

<asf pair>                       ::=  <asf type enumerated>
                                      <asf enumerated>

<asf type enumerated>            ::=  <LINE TYPE ASF>
                                    | <LINE WIDTH ASF>
                                    | <LINE COLOUR ASF>
                                    | <MARKER TYPE ASF>
                                    | <MARKER SIZE ASF>
                                    | <MARKER COLOUR ASF>
                                    | <TEXT FONT ASF>
                                    | <TEXT PRECISION ASF>
                                    | <CHARACTER EXPANSION FACTOR ASF>
                                    | <CHARACTER SPACING ASF>
                                    | <TEXT COLOUR ASF>
                                    | <INTERIOR STYLE ASF>
                                    | <FILL COLOUR ASF>
                                    | <HATCH INDEX ASF>
                                    | <PATTERN INDEX ASF>
                                    | <EDGE TYPE ASF>
                                    | <EDGE WIDTH ASF>
                                    | <EDGE COLOUR ASF>

<asf enumerated>                 ::=  <INDIVIDUAL>
                                    | <BUNDLED>
```

G.3.7 Escape elements

```
<escape element>                 ::=  <ESCAPE>
                                        <identifier>
                                        <data record>

<identifier>                     ::=  <integer>
```

G.3.8 External elements

```
<external element>               ::=  <MESSAGE>
                                        <action flag enumerated>
                                        <string>
                                    | <APPLICATION DATA>
                                        <integer>
                                        <data record>

<action flag enumerated>         ::=  <YES>
                                    | <NO>
```

G.4 Terminal symbols

The following are the terminals in this grammar. Their representation is dependent on the encoding scheme used. In annex A of the subsequent parts of this Standard, these encoding-dependent symbols are further described.

<element name>
<integer>
<real>
<vdc value>
<string>
<colour index>
<red green blue>
<integer precision value>
<real precision value>
<index precision value>
<colour precision value>
<colour index precision value>
<default colour precision indicator>
<vdc integer precision value>
<vdc real precision value>
<data record>

The CGM extended opcodes are encoding dependent. A complete list of them can be found in the productions for <element name enumerated> below.

The enumerated types are:

<INTEGER>
<REAL>
<ON>
<OFF>
<INDEXED>
<DIRECT>
<ABSTRACT>
<METRIC>
<ABSOLUTE>
<SCALED>
<94 CHAR>
<96 CHAR>
<MULTI-BYTE 94 CHAR>
<MULTI-BYTE 96 CHAR>
<COMPLETE CODE>
<BASIC 7-BIT>
<BASIC 8-BIT>
<EXTENDED 7-BIT>
<EXTENDED 8-BIT>
<INVISIBLE>
<VISIBLE>
<CLOSE INVISIBLE>
<CLOSE VISIBLE>
<PIE>
<CHORD>
<FINAL>
<NOT FINAL>
<INDIVIDUAL>
<BUNDLED>
<HOLLOW>
<SOLID>

<PATTERN>
<HATCH>
<EMPTY>
<STRING>
<CHARACTER>
<STROKE>
<LEFT>
<RIGHT>
<UP>
<DOWN>
<NORMAL HORIZONTAL>
<CENTRE>
<CONTINUOUS HORIZONTAL>
<NORMAL VERTICAL>
<TOP>
<CAP>
<HALF>
<BASE>
<BOTTOM>
<CONTINUOUS VERTICAL>
<YES>
<NO>
<LINE TYPE ASF>
<LINE WIDTH ASF>
<LINE COLOUR ASF>
<MARKER TYPE ASF>
<MARKER SIZE ASF>
<MARKER COLOUR ASF>
<TEXT FONT ASF>
<TEXT PRECISION ASF>
<CHARACTER EXPANSION FACTOR ASF>
<CHARACTER SPACING ASF>
<TEXT COLOUR ASF>
<INTERIOR STYLE ASF>
<HATCH INDEX ASF>
<PATTERN INDEX ASF>
<FILL COLOUR ASF>
<EDGE TYPE ASF>
<EDGE WIDTH ASF>
<EDGE COLOUR ASF>
<DRAWING SET>
<DRAWING PLUS CONTROL SET>

<element name enumerated> ::= <BEGIN METAFILE>
| <END METAFILE>
| <BEGIN PICTURE>
| <BEGIN PICTURE BODY>
| <END PICTURE>
| <METAFILE VERSION>
| <METAFILE DESCRIPTION>
| <VDC TYPE>
| <INTEGER PRECISION>
| <REAL PRECISION>
| <INDEX PRECISION>
| <COLOUR PRECISION>
| <COLOUR INDEX PRECISION>
| <MAXIMUM COLOUR INDEX>
| <COLOUR VALUE EXTENT>
| <METAFILE ELEMENT LIST>
| <METAFILE DEFAULTS REPLACEMENT>

| <FONT LIST>
| <CHARACTER SET LIST>
| <CHARACTER CODING ANNOUNCER>
| <SCALING MODE>
| <COLOUR SELECTION MODE>
| <LINE WIDTH SPECIFICATION MODE>
| <MARKER SIZE SPECIFICATION MODE>
| <EDGE WIDTH SPECIFICATION MODE>
| <VDC EXTENT>
| <BACKGROUND COLOUR>
| <VDC INTEGER PRECISION>
| <VDC REAL PRECISION>
| <AUXILIARY COLOUR>
| <TRANSPARENCY>
| <CLIP RECTANGLE>
| <CLIP INDICATOR>
| <POLYLINE>
| <DISJOINT POLYLINE>
| <POLYMARKER>
| <TEXT>
| <RESTRICTED TEXT>
| <APPEND TEXT>
| <POLYGON>
| <POLYGON SET>
| <CELL ARRAY>
| <GDP>
| <RECTANGLE>
| <CIRCLE>
| <CIRCULAR ARC 3 POINT>
| <CIRCULAR ARC 3 POINT CLOSE>
| <CIRCULAR ARC CENTRE>
| <CIRCULAR ARC CENTRE CLOSE>
| <ELLIPSE>
| <ELLIPTICAL ARC>
| <ELLIPTICAL ARC CLOSE>
| <LINE BUNDLE INDEX>
| <LINE TYPE>
| <LINE WIDTH>
| <LINE COLOUR>
| <MARKER BUNDLE INDEX>
| <MARKER TYPE>
| <MARKER SIZE>
| <MARKER COLOUR>
| <TEXT BUNDLE INDEX>
| <TEXT FONT INDEX>
| <TEXT PRECISION>
| <CHARACTER EXPANSION FACTOR>
| <CHARACTER SPACING>
| <TEXT COLOUR>
| <CHARACTER HEIGHT>
| <CHARACTER ORIENTATION>
| <TEXT PATH>
| <TEXT ALIGNMENT>
| <CHARACTER SET INDEX>
| <ALTERNATE CHARACTER SET INDEX>
| <FILL BUNDLE INDEX>
| <INTERIOR STYLE>
| <FILL COLOUR>
| <HATCH INDEX>
| <PATTERN INDEX>
| <EDGE BUNDLE INDEX>

| <EDGE TYPE>
| <EDGE WIDTH>
| <EDGE COLOUR>
| <EDGE VISIBILITY>
| <FILL REFERENCE POINT>
| <PATTERN TABLE>
| <PATTERN SIZE>
| <COLOUR TABLE>
| <ASPECT SOURCE FLAGS>
| <ESCAPE>
| <MESSAGE>
| <APPLICATION DATA>

Annex H

Relationship of CGM and GKS

(This annex does not form a part of the standard)

H.1 Introduction

The GKS Standard (ISO 7942) includes the concepts of metafile input (MI) and output (MO) workstations as well as functions providing access to and interpretation of metafiles. It does not, however, contain a metafile definition as part of the standard. Annex E of Part 1 of ISO/IEC 8632 provides a mapping to CGM version 1 metafiles.

This Annex provides a mapping between GKS and CGM version 2 metafiles.

H.2 Scope

The CGM version 2 captures static picture definitions. GKS provides many possibilities to generate images. This means that the strategies for generating picture definitions are numerous and complex. The best strategy to use in given circumstances is dictated by implementation and application requirements. This annex presents a detailed mapping between GKS and CGM only for one particular strategy.

The scope of this annex is further limited to generation of metafiles by GKS and interpretation of GKS-generated metafiles in GKS environments. There are many other scenarios for generation and interpretation of metafiles, such as interpretation by GKS of metafiles not generated by GKS and interpretation by non-GKS processes of GKS-generated metafiles. These scenarios are not dealt with in this annex. Annex C of this part of ISO/IEC 8632 presents context models dealing with such cases.

H.3 Overview of the differences between GKS and CGM version 2

While CGM supports all of the basic output functionality of GKS, a one-to-one mapping between GKS and CGM is not possible in all cases mainly because some CGM elements have no counterparts as GKS functions and some GKS functions have no corresponding CGM element. Examples of this are:

a) Delimiter elements, for example BEGIN PICTURE
b) Enhanced facilities for tailoring and controlling the interpretation of the metafile precision of various items, and the control of default values.
c) Extended capabilities in the area of text processing, for example named font, changing character sets and restricted text.

H.4 Mapping concepts

The tables in this annex present mappings between GKS and CGM elements.

H.4.1 Principles

The following principles are the basis of the GKS/CGM model of this annex and of the function mappings themselves:

a) conceptual compatibility with GKS;
b) compatibility with the design concepts of CGM;
c) extensibility of the elements taken from CGM to a GKS static picture-capture metafile.

H.4.2 Workstation

The CGM is generated, in this model, by a workstation of type MO. The behaviour of the workstation, particularly in response to dynamic GKS functions, can be illustrated by analogy: in most respects, the MO/CGM workstation in GKS may be implemented in a manner analogous to a workstation of category OUTPUT (for example, a plotter), whose device instruction set corresponds to the CGM elements. Strategies for correctly sending device instructions to such a real device are similar to those generating the proper elements on the metafile.

The CGM is read by a workstation of category MI. Certain elements, such as the metafile descriptor and precision-setting elements, are viewed as directives to the MI workstation itself, so that it may subsequently read the metafile contents correctly.

H.4.3 Picture generation

A metafile comprises a collection of mutually independent pictures. GKS does not have the concept of a "picture" as defined in CGM but it does formalize the notion of an empty view surface. GKS actions which cause clearing of the view surface, such as CLEAR WORKSTATION, are defined to delimit metafile pictures. There is another mechanism which leads to generation of pictures in this model of the GKS/CGM relationship. GKS contains functions which have potential dynamic effects on a non-empty display surface. The CGM design concepts exclude dynamic modification of pictures. For this reason all "dynamic modification accepted" values of a MO/CGM workstation will be conceptually IRG (implicit regeneration necessary).

The default value of the deferral state on an MO/CGM workstation is (ASTI-SUPPRESSED).

This model of the MO/CGM workstation defines that whenever a GKS function is invoked which causes a regeneration, then a new picture is output to the metafile.

H.4.4 Coordinates and clipping

The coordinate space of the metafile, VDC, is conceptually identical to the NDC space of GKS. The MAXIMUM VDC EXTENT element allows the mapping of VDC of either type (real or integer) on to the unit interval of NDC. Clipping and transformation are completely deferred to the metafile interpreter.

Clipping is always 'on' in the metafile, which is the default value of the CLIP INDICATOR element (hence CLIP INDICATOR elements need never be written to the metafile). The CGM CLIP RECTANGLE element has either the value of the 'clipping rectangle' entry of the GKS state list, or the MAXIMUM VDC EXTENT in VDC, depending upon whether the 'clipping indicator' entry in the GKS state list is 'clip' or 'noclip' respectively. Because the VDC EXTENT element always has the value of the GKS workstation window in VDC, the interpreter of the metafile has complete information to achieve GKS clipping.

H.4.5 Workstation transformation

The workstation transformation is defined in GKS by setting a workstation window in device-independent NDC and a workstation viewport in device-dependent DC. The workstation window is written to the metafile with the VDC EXTENT element. The workstation viewport is written to the metafile with the DEVICE VIEWPORT element.

The default values of DEVICE VIEWPORT MAPPING correspond to the GKS mapping of the device coordinate system onto the display space. The DEVICE VIEWPORT SPECIFICATION MODE is set to 'millimetres with scale factor' and metric scale factor 1000.0 within the METAFILE DEFAULTS REPLACEMENT element.

H.4.6 Metafile element list

The metafile element list short hand defined for use with GKS application is 'version 2 GKSM set'.

H.4.7 Relationship of fonts between CGM and GKS

The GKS standard includes the concepts of text output primitive attributes. However, the mechanism for specifying the text font differs from that specified in the CGM standard. This clause defines the approach to handling these attributes within the GKS environment.

H.4.7.1 Overview of the differences between GKS and CGM fonts

While CGM supports the TEXT output primitive attribute functionality of GKS, a one-to-one mapping between CGM and GKS is not possible in all cases. Specifically:

a) GKS and CGM differ in the way fonts are defined. In the CGM text fonts are defined with the FONT LIST element that associates font names or identifications with entries in a Font Table. In GKS, no mechanism is available for defining text fonts. GKS associates a unique text font number with each font. The Registration Authority is responsible for defining this mapping of font numbers to specific font identifications.

b) GKS and CGM differ in the way fonts are selected. In the CGM, text fonts are selected with the TEXT FONT INDEX element. The index selects an individual font from different fonts in the font list. In GKS, text fonts are selected with a font number. The font number selects a specific GKS registered font if the value is positive. If the font number is negative, an implementation dependent font is selected.

c) GKS and CGM differ on the independence of font and text precision. In the CGM, the font and text precision are specified by independent elements. In GKS, the font and text precision are specified together by a single function.

d) Some CGM Elements have no counterpart as GKS functions. These include auxiliary colour related elements, such as AUXILIARY COLOUR and TRANSPARENCY, that affect the presentation of text. This additional functionality of the CGM causes no special problems for a GKS environment interpreting a version 2 CGM.

e) The character set related elements CHARACTER SET LIST, CHARACTER CODING ANNOUNCER, CHARACTER SET INDEX, ALTERNATE CHARACTER SET INDEX have no counterpart in GKS. GKS does not recognize the concept of character set as a separate concept from the font concept. GKS implementors are encouraged to provide a mapping to the character set elements for both MO and MI workstations to increase the possibility of transferring metafiles between GKS environments and other systems.

H.4.7.2 Suggestion for interpretation of CGM font information by GKS

GKS environments interpreting a CGM specify fonts with a font number. It is assumed that GKS maintains a list associating positive font numbers with a GKS registered font name or identifier. Private font numbers (i.e. negative values) must be maintained in an implementation-dependent list of associations. As the FONT LIST element is interpreted, an additional list must be maintained that associates individual font names specified in the CGM with a font index. When the TEXT FONT INDEX element is interpreted, the font name associated with the font index is determined from the list of currently used fonts. The font name is used to determine the GKS font number associated with this font from a list of GKS registered fonts. This font number is used as the font parameter of the TEXT FONT AND PRECISION function. The value of the precision parameter is taken from the TEXT PRECISION element.

H.4.7.3 Generating CGM font information from GKS

When generating font information from GKS via TEXT FONT AND PRECISION it is recommended that the generator also writes the elements CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX as well as TEXT FONT INDEX and TEXT PRECISION. The generator is assumed to have a table associating the positive font numbers of GKS with the registered names. The generator shall put a FONT LIST element in the Metafile Descriptor with the names of those fonts referenced by positive GKS font numbers. Negative GKS font numbers are private and must be mapped to CGM font indices which are positive.

H.5 Metafile generation

Included in following tables is a particular set of mappings of the GKS function, workstation state list entries and segment state list entries onto CGM elements. The mappings presented are deemed usable and suitable for guiding implementation of a CGM picture generator in a GKS environment. The mapping concepts of H.4 are assumed.

H.5.1 Control functions

Table 11 - Mapping of control functions.

GKS function	CGM version 2 elements	Notes
OPEN WORKSTATION	BEGIN METAFILE (Metafile Descriptor) BEGIN PICTURE store current workstation state list BEGIN PICTURE BODY	(1) (2) (3) (4)
CLOSE WORKSTATION	END PICTURE END METAFILE	
ACTIVATE WORKSTATION	attribute settings CLIP RECTANGLE enable output to metafile	(5) (6)
DEACTIVATE WORKSTATION	disable output to metafile	
CLEAR WORKSTATION control flag = CONDITIONAL display space empty = EMPTY	no Action	
CLEAR WORKSTATION display space empty = NOTEMPTY	END PICTURE BEGIN PICTURE store current workstation state list BEGIN PICTURE BODY attribute settings CLIP RECTANGLE	 (3) (4) (5) (6)
REDRAW ALL SEGMENTS ON WORKSTATION display space empty = EMPTY	no Action	
REDRAW ALL SEGMENTS ON WORKSTATION display space empty = NOTEMPTY	END PICTURE BEGIN PICTURE store current workstation state list BEGIN PICTURE BODY attribute settings CLIP RECTANGLE generate all visible segments stored for the MO workstation	 (3) (4) (5) (6) (7)
UPDATE WORKSTATION regeneration flag = PERFORM new frame action necessary at update = YES	as REDRAW ALL SEGMENTS ON WORKSTATION	
UPDATE WORKSTATION regeneration flag = PERFORM new frame action necessary at update = NO or UPDATE WORKSTATION regeneration flag = POSTPONE	no Action	
SET DEFERRAL STATE new frame action necessary at update = YES	as REDRAW ALL SEGMENTS ON WORKSTATION	
ESCAPE	ESCAPE	
MESSAGE	MESSAGE	(8)

NOTES

1 The use of the identifier parameter in BEGIN METAFILE is implementation dependent.
2 See H.5.5, Metafile Description
3 The use of the identifier parameter in BEGIN PICTURE is implementation dependent.
4 See H.5.6, Mapping of workstation state list entries to CGM elements.
5 The attribute settings ensure that the metafile attributes in effect when the first graphical primitive element of a picture is encountered match the current attributes from the GKS state list.
6 A CLIP RECTANGLE is written to the metafile with the values of the MAXIMUM VDC EXTENT if the 'clipping indicator' entry in the GKS state list is 'noclip', or with values corresponding to the 'clipping rectangle' in the GKS state list if the 'clipping indicator' entry in the GKS state list is 'clip'.
7 Generate a sequence of CGM-elements for every segment as ASSOCIATE SEGMENT WITH WORKSTATION (see H.5.3.4)
8 Action required flag is set to 'noaction'.

H.5.2 GKS functions leading to an implicit regeneration

Depending on the deferral state the following GKS functions may act as REDRAW ALL SEGMENTS ON WORKSTATION because conceptually all corresponding "dynamic modification accepted" entries in the workstation description table are set to IRG (see H.4.3):

SET POLYLINE REPRESENTATION
SET MARKER REPRESENTATION
SET TEXT REPRESENTATION
SET INTERIOR REPRESENTATION
SET PATTERN REPRESENTATION
SET COLOUR REPRESENTATION

SET WORKSTATION WINDOW
SET WORKSTATION VIEWPORT

SET SEGMENT TRANSFORMATION
SET VISIBILITY
SET HIGHLIGHTING
SET SEGMENT PRIORITY

NOTE - all primitives added to open segments overlapping segments of higher priority

DELETE SEGMENT
DELETE SEGMENT FROM WORKSTATION
ASSOCIATE SEGMENT WITH WORKSTATION

H.5.3 GKS functions with no direct dynamic effect

H.5.3.1 Output functions

Table 12 - Mapping of output functions.

GKS function	CGM Element	Notes
POLYLINE	POLYLINE	
POLYMARKER	POLYMARKER	
TEXT	TEXT	(1)
FILL AREA	POLYGON	
CELL ARRAY	CELL ARRAY	(2)
GDP	GDP	

NOTES

1 The text flag is set to 'final'.
2 The default colour selection mode 'indexed' applies.

H.5.3.2 Attributes

Table 13 - Mapping of attribute functions.

GKS function	CGM element	Notes
SET POLYLINE INDEX	LINE BUNDLE INDEX	
SET LINETYPE	LINE TYPE	
SET LINEWIDTH SCALE FACTOR	LINE WIDTH	(1)
SET POLYLINE COLOUR INDEX	LINE COLOUR	(2)
SET POLYMARKER INDEX	MARKER BUNDLE INDEX	
SET MARKER TYPE	MARKER TYPE	
SET MARKER SIZE SCALE FACTOR	MARKER SIZE	(1)
SET POLYMARKER COLOUR INDEX	MARKER COLOUR	(2)
SET TEXT INDEX	TEXT BUNDLE INDEX	
SET TEXT FONT AND PRECISION	TEXT FONT INDEX	(3)
	TEXT PRECISION	
	CHARACTER SET INDEX	
	ALTERNATE CHARACTER SET INDEX	
SET CHARACTER EXPANSION FACTOR	CHARACTER EXPANSION FACTOR	
SET CHARACTER SPACING	CHARACTER SPACING	
SET TEXT COLOUR INDEX	TEXT COLOUR	(2)
SET CHARACTER HEIGHT	CHARACTER HEIGHT	
SET CHARACTER UP VECTOR	CHARACTER ORIENTATION	
SET TEXT PATH	TEXT PATH	
SET TEXT ALIGNMENT	TEXT ALIGNMENT	
SET FILL AREA INDEX	FILL BUNDLE INDEX	
SET FILL AREA STYLE	INTERIOR STYLE	
SET FILL AREA STYPE INDEX	HATCH INDEX	(4)
	PATTERN INDEX	
SET FILL AREA COLOUR INDEX	FILL COLOUR	
SET PATTERN SIZE	PATTERN SIZE	
SET PATTERN REFERENCE POINT	FILL REFERENCE POINT	
SET ASPECT SOURCE FLAG	ASPECT SOURCE FLAGS	
SET PICK IDENTIFIER	PICK IDENTIFIER	

NOTES

1 The default specification modes 'scaled' apply.

2 The default colour selection mode 'indexed' applies.

3 GKS includes the notion of character set within 'font', whereas CGM separates the two concepts. When the value of 'font' in the GKS state list changes, then the CGM elements TEXT FONT INDEX, TEXT PRECISION, CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX are written to the metafile, each with the value of the 'font' and 'precision' entry in the GKS state list. The CGM font index is determined as described in H.4.7.3. The elements shall appear consecutively in the metafile but may appear in any order.

4 Legal values of the GKS 'fill area style index' differ depending upon whether the current interior style is 'hatch' or 'pattern'. Therefore a negative GKS style index results only in the generation of the HATCH INDEX element, and a positive value results in the generation of both the HATCH INDEX and PATTERN INDEX elements.

H.5.3.3 Transformation functions

Table 14 - Mapping of transformation functions.

GKS function	CGM Element	Notes
SET WINDOW (of current selected normalisation transformation)	CHARACTER HEIGHT CHARACTER ORIENTATION PATTERN SIZE FILL REFERENCE POINT	
SET VIEWPORT (of current selected normalisation transformation)	CHARACTER HEIGHT CHARACTER ORIENTATION PATTERN SIZE FILL REFERENCE POINT CLIP RECTANGLE	 (1)
SELECT NORMALISATION TRANSFORMATION	CHARACTER HEIGHT CHARACTER ORIENTATION PATTERN SIZE FILL REFERENCE POINT CLIP RECTANGLE	 (1)
SET CLIP INDICATOR	CLIP RECTANGLE	(2)

NOTES

1 If the 'clipping rectangle' entry in the GKS state list is changed, then a CLIP RECTANGLE element is written to the metafile. The element is written with the values of MAXIMUM VDC EXTENT if the 'clipping indicator' entry in the GKS state list is 'noclip', or with values corresponding to the 'clipping rectangle' in the GKS state list if the 'clipping indicator' entry in the GKS list is 'clip'.

2 If the 'clipping indicator' entry in the GKS state list is changed, then a CLIP RECTANGLE element is written to the metafile. The element is written with the values of MAXIMUM VDC EXTENT if the 'clipping indicator' entry in the GKS state list is changed to 'noclip', or with values corresponding to the 'clipping rectangle' in the GKS state list if the 'clipping indicator' entry in the GKS state list is changed to 'clip'.

H.5.3.4 Segment manipulation functions

Table 15 - Mapping of segment manipulation functions

GKS function	CGM element	Notes
CREATE SEGMENT	BEGIN SEGMENT	
CLOSE SEGMENT	END SEGMENT	
RENAME SEGMENT	no action	
ASSOCIATE SEGMENT WITH WORKSTATION	BEGIN SEGMENT (segment attributes from the segment state list) (primitives and their associated attributes and clip rectangle) END SEGMENT	 (1)
COPY SEGMENT TO WORKSTATION	(transformed primitives and their associated attributes and clip rectangle)	(1) (2)
INSERT SEGMENT TO WORKSTATION	(transformed primitives and their associated attributes and clip rectangle)	(3,4)

NOTES

1 The associated clip rectangle.
2 Primitives transformed by the segment transformation.
3 Primitives transformed by the segment transformation followed by the insert transformation.
4 A clip rectangle corresponding to the clipping rectangle in the GKS state list if the 'clipping indicator' entry in the GKS state list is 'clip', or the corresponding [0,1]x[0,1] clip rectangle - which is the maximum VDC extent - if the 'clipping indicator' entry in the GKS state list is 'noclip'.

H.5.3.5 Mappings of segment attributes

See H.5.3.4 and H.5.7.

H.5.4 GKS function with no action

The following GKS functions have no action when the Metafile Output workstation is writing a CGM:

SET DETECTABLITY
SET VIEWPORT INPUT PRIORITY
all input function
all inquiry function
all utility function
all error handling function

H.5.5 Metafile Description

At the beginning of a metafile is a set of Metafile Descriptor (MD) elements. It is useful to view these elements as forming a Metafile Description Table (similar to the GKS and Workstation Description Table in GKS).

In the GKS context, the description table shown in table 16 would be written at the beginning of a metafile. For the elements which are listed as "i.d.", it is implementation-dependent both whether the elements are included in the table - except for the mandatory elements - and what values are assigned to the elements if they are written to the metafile.

Table 16 - The description table for the CGM written by a GKS program.

Metafile element list	Element Value	Mandatory
METAFILE VERSION	2	X
METAFILE ELEMENT LIST	Elements of the Version 2 GKSM set	X
METAFILE DESCRIPTION	i.d.	
VDC TYPE	i.d.	
INTEGER PRECISION	i.d.	
REAL PRECISION	i.d.	
INDEX PRECISION	i.d.	
COLOUR PRECISION	i.d.	
MAXIMUM COLOUR INDEX	id	
COLOUR INDEX PRECISION	i.d.	
COLOUR VALUE EXTENT	i.d.	
METAFILE DEFAULT REPLACEMENT	i.d.	
FONT LIST	i.d.	
CHARACTER CODING ANNOUNCER	i.d.	
CHARACTER SET LIST	i.d.	
NAME PRECISION	i.d.	
MAXIMUM VDC EXTENT	i.d.	
SEGMENT PRIORITY EXTENT	i.d.	

i.d. = implementation dependent

METAFILE VERSION and METAFILE ELEMENT LIST are mandatory. All metafile defaults satisfy the GKS Description Table. Inclusion of the METAFILE DEFAULTS REPLACEMENT element to change any control, picture descriptor, and attribute defaults is optional and implementation dependent.

H.5.6 Workstation state list entries

Table 17 - Mapping of workstation state list entries.

GKS workstation state list entry	CGM element	Notes
requested workstation window	VDC EXTENT	(1)
requested workstation viewport	DEVICE VIEWPORT	(2)
every entry of polyline bundle table	LINE REPRESENTATION	
every entry of polymarker bundle table	MARKER REPRESENTATION	
every entry of text bundle table	TEXT REPRESENTATION	
every entry of interior bundle table	FILL REPRESENTATION	
every entry of pattern table	PATTERN TABLE	
every entry of colour table	COLOUR TABLE	

NOTES

1 The position of the workstation window within the NDC unit square corresponds to the position of the VDC extent within the maximum VDC extent.

2 DEVICE VIEWPORT SPECIFICATION MODE and DEVICE VIEWPORT MAPPING may be specified only within METAFILE DEFAULTS REPLACEMENT in the metafile descriptor. The VC specifier may be either 'millimetres with scale factor' with metric scale factor 1000.0, or 'physical device coordinates'.

H.5.7 Segment state list entries

Table 18 - Mapping of segment state list entries.

GKS segment state list entry	CGM element	Notes
segment transformation matrix	SEGMENT TRANSFORMATION	
visibility	-	(1)
highlighting	SEGMENT HIGHLIGHTING	
segment priority	SEGMENT DISPLAY PRIORITY	
	SEGMENT PICK PRIORITY	(2)
detectability	-	

NOTES

1 invisible segments are not mapped.

2 The elements shall appear consecutively in the metafile but may appear in any order.

H.5.8 Metafile function

Table 19 - Mapping of the metafile function.

GKS function	CGM element	Notes
WRITE ITEM TO METAFILE	APPLICATION DATA	(1)

NOTES

1 The GKS item type is mapped to the CGM application data identifier.

H.6 Metafile interpretation

This sub-clause describes how metafile elements from a version 2 metafile generated by a GKS program according to the mapping described in sub-clause H.5 are subsequently interpreted by the GKS INTERPRET ITEM function and/or the MI/CGM workstation. Other guidelines for interpretation are possible.

Those CGM elements that do not map to a GKS item are viewed as directives to the MI/CGM workstation itself, so that it may correctly read the metafile contents.

A number of the elements below are specified as causing GKS state list entries to be set and have parameters specified in VDC (which corresponds to GKS NDC). The GKS state list entries are in WC. The VDC (NDC) are mapped by the inverse of the current normalization transformation before the GKS state list values are set. The table also includes item types to be returned to GKS. These are adopted from GKS Annex E.

H.6.1 Delimiter elements

Table 20 - Mapping of delimiter elements.

CGM Element	GKS Metafile Interface	Item	Notes
BEGIN METAFILE	-	-	(1)
END METAFILE	END ITEM	0	(2)
BEGIN PICTURE	-	-	(3)
BEGIN PICTURE BODY	CLEAR WORKSTATION	1	(4)
END PICTURE	-	-	
BEGIN SEGMENT	CREATE SEGMENT	81	
END SEGMENT	CLOSE SEGMENT	82	

NOTES

1 The first CGM element interpreted by the MI workstation. The metafile description table immediately follows. Its elements inform the MI workstation how to read the metafile.

2 No further items may be read.

3 Appropriate GKS state list values are set to correspond to CGM defaults. Appropriate workstation state list values on active OUTPUT and OUTIN workstations are set to correspond to CGM defaults. It is not intended that this action, or the interpretation of any picture descriptor elements, cause any immediate dynamic changes to the view surface, which is cleared upon BEGIN PICTURE BODY - the implementation may wish to buffer these actions to suppress such changes, if such changes are undesirable. Only picture descriptor elements may be interpreted until BEGIN PICTURE BODY.

4 Causes a CLEAR WORKSTATION on all active workstations.

H.6.2 Metafile descriptor elements

All elements in this class contain only directives to the MI workstation, their interpretation does not correspond to the invocation of any GKS function.

Table 21 - Mapping of metafile descriptor elements.

CGM Element	GKS Metafile Interface	Item	Notes
METAFILE VERSION	-	-	(1)
METAFILE DESCRIPTION	-	-	
VDC TYPE	-	-	
INTEGER PRECISION	-	-	
REAL PRECISION	-	-	
INDEX PRECISION	-	-	
COLOUR PRECISION	-	-	
COLOUR INDEX PRECISION	-	-	
MAXIMUM COLOUR INDEX	-	-	
COLOUR VALUE EXTENT	-	-	(2)
METAFILE ELEMENT LIST	-	-	
METAFILE DEFAULTS REPLACEMENT	-	-	
FONT LIST	-	-	
CHARACTER SET LIST	-	-	
CHARACTER CODING ANNOUNCER	-	-	
NAME PRECISION	-	-	
MAXIMUM VDC EXTENT	-	-	(3)
SEGMENT PRIORITY EXTENT	-	-	(4)

NOTES

1 The value of the parameter must be 2.
2 Used to normalize colour direct values to the continuous range of real numbers [0,1].
3 Used to normalize VDC range (i.e.NDC) and applies to VDC type INTEGER or REAL
4 Used to normalize segment priority to the continuous range of real numbers [0,1].

H.6.3 Picture descriptor elements

Table 22 - Mapping of picture descriptor elements.

CGM Element	GKS Metafile Interface	Item	Notes
VDC EXTENT	WOKSTATION WINDOW	71	
DEVICE VIEWPORT	WORKSTATION VIEWPORT	72	
DEVICE VIEWPORT SPECIFICATION MODE	-	-	(1)
DEVICE VIEWPORT MAPPING	-	-	(2)
LINE REPRESENTATION	POLYLINE REPRESENTATION	51	
MARKER REPRESENTATION	POLYMARKER REPRESENTATION	52	
TEXT REPRESENTATION	TEXT REPRESENTATION	53	
FILL REPRESENTATION	FILL AREA REPRESENTATION 54		
PATTERN TABLE	PATTERN REPRESENTATION	56	
COLOUR TABLE	COLOUR REPRESENTATION	57	

NOTES

1 The VC specifier may be either 'millimetres with scale factor' with metric scale factor equal to 1000.0 or 'physical device coordinates'. DEVICE VIEWPORT SPECIFICATION MODE may occur only within METAFILE DEFAULTS REPLACEMENT.
2 The isotropy flag must be 'forced' and the alignment flags must be 'left' and 'bottom'. DEVICE VIEWPORT MAPPING may occur only within METAFILE DEFAULTS REPLACEMENT.

H.6.4 Control elements

Table 23 - Mapping of control elements.

CGM Element	GKS Metafile Interface	Item	Notes
VDC INTEGER PRECISION	-	-	
VDC REAL PRECISION	-	-	
CLIP RECTANGLE	CLIPPING RECTANGLE	61	

H.6.5 Graphical primitive elements

Table 24 - Mapping of graphical primitive elements.

CGM Element	GKS Metafile Interface	Item	Notes
POLYLINE	POLYLINE	11	
POLYMARKER	POLYMARKER	12	
TEXT	TEXT	13	(1)
POLYGON	FILL AREA	14	
CELL ARRAY	CELL ARRAY	15	
GDP	GDP	16	

NOTES

1 The text flag should be 'final'.

H.6.6 Attribute elements

Table 25 - Mapping of attribute elements.

CGM Element	GKS Metafile Interface	Item	Notes
LINE BUNDLE INDEX	POLYLINE INDEX	21	
LINE TYPE	LINE TYPE	22	
LINE WIDTH	LINE WIDTH SCALE FACTOR	23	(1)
LINE COLOUR	POLYLINE COLOUR INDEX	24	(2)
MARKER BUNDLE INDEX	POLYMARKER INDEX	25	
MARKER TYPE	MARKER TYPE	26	
MARKER SIZE	MARKER SIZE SCALE FACT	27	(1)
MARKER COLOUR	POLYMARKER COLOUR INDEX	28	(2)
TEXT BUNDLE INDEX	TEXT INDEX	29	
TEXT FONT INDEX	TEXT FONT AND PRECISION	30	(3)
TEXT PRECISION	TEXT FONT AND PRECISION	30	(3)
CHARACTER EXPANSION FACTOR	CHARACTER EXPANSION FACTOR	31	
CHARACTER SPACING	CHARACTER SPACING	32	
TEXT COLOUR	TEXT COLOUR INDEX	33	(2)
CHARACTER HEIGHT	CHARACTER VECTORS	34	(4)
CHARACTER ORIENTATION	CHARACTER VECTORS	34	(4)
TEXT PATH	TEXT PATH	35	
TEXT ALIGNMENT	TEXT ALIGNMENT	36	
CHARACTER SET INDEX	TEXT FONT AND PRECISION	30	(3)
ALTERNATE CHARACTER SET INDEX	TEXT FONT AND PRECISION	30	(3)
FILL BUNDLE INDEX	FILL AREA INDEX	37	
INTERIOR STYLE	FILL AREA INTERIOR STYLE	38	
FILL COLOUR	FILL AREA COLOUR INDEX	40	(2)
HATCH INDEX	FILL AREA STYLE INDEX	39	
PATTERN INDEX	FILL AREA STYLE INDEX	39	
PATTERN SIZE	PATTERN VECTORS	41	
FILL REFERENCE POINT	PATTERN REFERENCE POINT	42	
ASPECT SOURCE FLAGS	ASPECT SOURCE FLAGS	43	(5)
PICK IDENTIFIER	PICK IDENTIFIER	44	

NOTES

1 The default specification modes 'scaled' applies.

2 The default colour selection mode 'indexed' applies.

3 Four CGM elements supply the relevant parameter values of the GKS TEXT FONT AND PRECISION item (either explicitly or implicitly by default): TEXT FONT INDEX, TEXT PRECISION, CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX. The corresponding GKS font number may be determined as described in sub-clause H.4.7.2. The occurrence of only one of the four CGM elements uniquely indicates the mapping to GKS TEXT FONT AND PRECISION. The occurrence of more than one CGM element within one sequence in any order causes the corresponding GKS item to be returned once.

4 Two CGM elements supply the relevant parameter values of the GKS CHARACTER VECTORS item (either explicitly or implicitly by default) : CHARACTER HEIGHT and CHARACTER ORIENTATION. The occurrence of only one of the two CGM elements uniquely indicates the mapping to GKS CHARACTER VECTORS. The occurrence of the two CGM elements within one sequence in any order causes the corresponding GKS item to be returned once.

5 TEXT FONT ASF and TEXT PRECISION ASF are equal; they correspond to GKS TEXT FONT AND PRECISION ASF. HATCH INDEX and PATTERN INDEX ASF are equal; they correspond to GKS FILL AREA STYLE INDEX ASF.

H.6.7 Escape and external elements

Table 26 - Mapping of escape and external elements.

CGM Element	GKS Metafile Interface	Item	Notes
ESCAPE	ESCAPE	6	
MESSAGE	MESSAGE	5	(1)
APPLICATION DATA	USER ITEM	>100	

NOTES

1 The 'action required' flag should be 'no action'.

H.6.8 Mapping of segment attribute elements

Table 27 - Mapping of segment attribute elements.

CGM Element	GKS Metafile Interface	Item	Notes
SEGMENT TRANSFORMATION	SET SEGMENT TRANSFORMATION	91	
SEGMENT HIGHLIGHTING	SET HIGHLIGHTING	93	
SEGMENT DISPLAY PRIORITY	SET SEGMENT PRIORITY	94	(1)
SEGMENT PICK PRIORITY	SET SEGMENT PRIORITY	94	(1)

NOTES

1 Both CGM SEGMENT DISPLAY PRIORITY and SEGMENT PICK PRIORITY supply the parameter value of the GKS SET SEGMENT PRIORITY item.

ISO/IEC 8632-2 : 1987/Am.1 : 1990

Information processing systems - Computer graphics - Metafile for the storage and transfer of picture descriptive information -

Part 2: Character Encoding

Amendment 1

Page 10

Add the following to the end of 5.3:

3/8 for Segment Control Elements and Segment Attribute Elements

Page 11

Add the following to table 1:

opcode	7 bit coding		8 bit coding	
BEGIN SEGMENT opcode	3/0	2/5	03/0	02/5
END SEGMENT opcode	3/0	2/6	03/0	02/6
BEGIN FIGURE opcode	3/0	2/7	03/0	02/7
END FIGURE opcode	3/0	2/8	03/0	02/8
NAME PRECISION opcode	3/1	3/0	03/1	03/0
MAXIMUM VDC EXTENT opcode	3/1	3/1	03/1	03/1
SEGMENT PRIORITY EXTENT opcode	3/1	3/2	03/1	03/2
DEVICE VIEWPORT opcode	3/2	2/7	03/2	02/7
DEVICE VIEWPORT SPEC. MODE opcode	3/2	2/8	03/2	02/8
DEVICE VIEWPORT MAPPING opcode	3/2	2/9	03/2	02/9
LINE REPRESENTATION opcode	3/2	2/10	03/2	02/10
MARKER REPRESENTATION opcode	3/2	2/11	03/2	02/11
TEXT REPRESENTATION opcode	3/2	2/12	03/2	02/12
FILL REPRESENTATION opcode	3/2	2/13	03/2	02/13
EDGE REPRESENTATION opcode	3/2	2/14	03/2	02/14
LINE CLIPPING MODE opcode	3/3	2/6	03/3	02/6
MARKER CLIPPING MODE opcode	3/3	2/7	03/3	02/7
EDGE CLIPPING MODE opcode	3/3	2/8	03/3	02/8
NEW REGION opcode	3/3	2/9	03/3	02/9
SAVE PRIMITIVE CONTEXT opcode	3/3	2/10	03/3	02/10
RESTORE PRIMITIVE CONTEXT opcode	3/3	2/11	03/3	02/11
CIRCULAR ARC CENTRE REVERSED opcode	3/4	2/8	03/4	02/8
CONNECTING EDGE opcode	3/4	2/9	03/4	02/9
PICK IDENTIFIER opcode	3/6	3/2	03/6	03/2
COPY SEGMENT opcode	3/8	2/0	03/8	02/0
INHERITANCE FILTER opcode	3/8	2/1	03/8	02/1
CLIP INHERITANCE opcode	3/8	2/2	03/8	02/2
SEGMENT TRANSFORMATION opcode	3/8	2/3	03/8	02/3
SEGMENT HIGHLIGHTING opcode	3/8	2/4	03/8	02/4
SEGMENT DISPLAY PRIORITY opcode	3/8	2/5	03/8	02/5
SEGMENT PICK PRIORITY opcode	3/8	2/6	03/8	02/6

Page 27

Add the following after 6.12:

6.13 Coding VCs and viewport point parameters

A viewport point (VP) is a pair of VC (Viewport Coordinate) scalars representing the x and y coordinates of a point in viewport specification space. A VC scalar is either an integer or real number according to whether VIEWPORT SPECIFICATION MODE is 'fraction of display surface', 'millimetres with scale factor' or 'physical device coordinates'.

When VIEWPORT SPECIFICATION MODE is 'fraction of display surface', the encoding of the VC and viewport point data type is as described in 6.4, Coding Real Numbers. The size of the viewport point parameters is limited by the current REAL PRECISION value.

When VIEWPORT SPECIFICATION MODE is 'millimetres with scale factor' or 'physical device coordintates', the encoding of the viewport point data type is as described in 6.3, Coding Integers. The size of the viewport point parameters is limited by the current INTEGER PRECISION value.

6.14 Name parameters

Name parameters are coded as integers (basic format) at NAME PRECISION.

Page 31

Add the following after 8.1.5:

8.1.6 BEGIN SEGMENT

<BEGIN-SEGMENT-opcode: 3/0 2/5>
<name: segment-identifier>

<name: segment-identifier> = <integer>

8.1.7 END SEGMENT

<END-SEGMENT-opcode: 3/0 2/6>

8.1.8 BEGIN FIGURE

<BEGIN-FIGURE-opcode: 3/0 2/7>

8.1.9 END FIGURE

<END-FIGURE-opcode: 3/0 2/8>

Page 34

Add the following to the <enumerated: element set> of 8.2.11:

|<integer:2> {VERSION 2 SET}
|<integer:3> {EXTENDED PRIMITIVES SET}
|<integer:4> {VERSION 2 GKSM SET}

Page 36

Add the following after 8.2.15:

8.2.16 NAME PRECISION

<NAME-PRECISION-opcode: 3/1 3/0>
<integer: largest-name-code + 1>

The largest-name-code indicates how many bits occur in the largest possible magnitude for a name.

8.2.17 MAXIMUM VDC EXTENT

<MAXIMUM-VDC-EXTENT-opcode: 3/1 3/1>
<point: first-corner>
<point: second-corner>

8.2.18 SEGMENT PRIORITY EXTENT

< SEGMENT-PRIORITY-EXTENT-opcode: 3/1 3/2>
<integer: minimum-segment-priority-value>
<integer: maximum-segment-priority-value>

Page 38

Add the following after 8.3.7:

8.3.8 DEVICE VIEWPORT

<DEVICE-VIEWPORT-opcode: 3/2 2/7>
<viewport point: first-corner>
<viewport point: second-corner>

8.3.9 DEVICE VIEWPORT SPECIFICATION MODE

<DEVICE-VIEWPORT-SPECIFICATION-MODE-opcode: 3/2 2/8>
<enumerated: VC-specifier>
<real: metric-scale-factor>

<enumerated: VC-specifier>	=	<integer:0>	{fraction of display surface}
	|	<integer:1>	{mm with scale factor}
	|	<integer:2>	{physical device coordinates}

8.3.10 DEVICE VIEWPORT MAPPING

<DEVICE-VIEWPORT-MAPPING-opcode: 3/2 2/9>
<enumerated: isotropy-flag>
<enumerated: horizontal-alignment-flag>
<enumerated: vertical-alignment-flag>

<enumerated: isotropy-flag>	=	<integer:0>	{not forced}
	|	<integer:1>	{forced}
<enumerated: horizontal-alignment-flag>	=	<integer:0>	{left}
	|	<integer:1>	{centre}
	|	<integer:2>	{right}
<enumerated: vertical-alignment-flag>	=	<integer:0>	{bottom}
	|	<integer:1>	{centre}
	|	<integer:2>	{top}

8.3.11 LINE REPRESENTATION

<LINE-REPRESENTATION-opcode: 3/2 2/10>
<index: line-bundle-index>
<index: line-type>
<line-width-specifier>
<colour-specifier>

<index: line-bundle-index>	=	<positive integer>	
<index: line-type>	=	<integer: 1>	{solid}
	|	<integer: 2>	{dash}
	|	<integer: 3>	{dot}
	|	<integer: 4>	{dash-dot}
	|	<integer: 5>	{dash-dot-dot}
	|	<integer: negative>	{private line type}
<line-width-specifier>	=	<real: line width scale factor> {if LINE WIDTH SPECIFICATION MODE is scaled}	
	|	<VDC: line width> {if LINE WIDTH SPECIFICATION MODE is absolute}	
<colour-specifier>	=	<integer: colour index> {if COLOUR SELECTION MODE is indexed}	
	|	<RGB> {if COLOUR SELECTION MODE is direct}	
<integer: colour-index>	=	<non-negative integer>	

NOTE - Line types with values above 5 are reserved for registration.

8.3.12 MARKER REPRESENTATION

<MARKER-REPRESENTATION-opcode: 3/2 2/11>
<index: marker-bundle-index>
<index: marker-type>
<marker-size-specifier>
<colour-specifier>

<index: marker-bundle-index>	=	<positive integer>	
<index: marker-type>	=	<integer: 1>	{dot}
	|	<integer: 2>	{plus}
	|	<integer: 3>	{asterisk}
	|	<integer: 4>	{circle}
	|	<integer: 5>	{cross}
	|	<integer: negative>	{private marker type}
<marker-size-specifier>	=	<real: marker size scale factor> {if MARKER SIZE SPECIFICATION MODE is scaled}	
	|	<VDC: marker size> {if MARKER-SIZE SPECIFICATION MODE is absolute}	
<colour-specifier>	=	<integer: colour index> {if COLOUR SELECTION MODE is indexed}	
	|	<RGB> {if COLOUR SELECTION MODE is direct}	
<integer: colour-index>	=	<non-negative integer>	

NOTE - Marker types with values above 5 are reserved for registration.

8.3.13 TEXT REPRESENTATION

<TEXT-REPRESENTATION-opcode: 3/2 2/12>
<index: text-bundle-index>
<integer: text-font-index>
<enumerated: text-precision>
<real: character-spacing>
<real: expansion-factor>
<colour-specifier>

<index: text-bundle-index>	=	<positive integer>	
<integer: text-font-index>	=	<positive integer>	
<enumerated: text-precision>	=	<integer:0>	{string}
	|	<integer:1>	{character}
	|	<integer:2>	{stroke}
<real: character spacing>	=	<real>	
<real: expansion-factor>	=	<non-negative real>	
<colour-specifier>	=	<integer: colour index> {if COLOUR SELECTION MODE is indexed}	
	|	<RGB> {if COLOUR SELECTION MODE is direct}	
<integer: colour-index>	=	<non-negative integer>	

8.3.14 FILL REPRESENTATION

<FILL-REPRESENTATION-opcode: 3/2 2/13>
<index: fill-bundle-index>
<enumerated: interior-style>
<colour-specifier>
<index: hatch-index>
<index: pattern-index>

<index:fill-bundle-index>	=	<positive integer>	
<enumerated: interior-style>	=	<integer:0>	{hollow}
	|	<integer:1>	{solid}
	|	<integer:2>	{pattern}
	|	<integer:3>	{hatch}
	|	<integer:4>	{empty}
	|	<integer:negative>	{private style}
<colour-specifier>	=	<integer:colour index> {if COLOUR SELECTION MODE is indexed}	
	|	<RGB> {if COLOUR SELECTION MODE is direct}	
<index: hatch-index>	=	<integer:1>	{horizontal}
	|	<integer:2>	{vertical}
	|	<integer:3>	{positive slope}
	|	<integer:4>	{negative slope}
	|	<integer:5>	{horizontal/vertical cross}
	|	<integer:6>	{positive/negative cross}
	|	<integer:negative>	{private styles}
<index: pattern-index>	=	<positive integer>	
<integer: colour index>	=	<non-negative integer>	

NOTE - Hatch indices with values above 6 are reserved for registration.

8.3.15 EDGE REPRESENTATION

<EDGE-REPRESENTATION-opcode: 3/2 2/14>
<index: edge-bundle-index>
<index: edge-type>
<edge-width-specifier>
<colour-specifier>

<index: edge-bundle-index>	=	<positive integer>	
<index: edge-type>	=	<integer: 1>	{solid}
	|	<integer: 2>	{dash}
	|	<integer: 3>	{dot}
	|	<integer: 4>	{dash-dot}
	|	<integer: 5>	{dash-dot-dot}
	|	<integer: negative>	{private edge type}
<edge-width-specifier>	=	<real: edge width scale factor> {if EDGE WIDTH SPECIFICATION MODE is scaled}	
	|	<VDC: edge width> {if EDGE WIDTH SPECIFICATION MODE is absolute}	
<colour-specifier>	=	<integer: colour-index> {if COLOUR SELECTION MODE is indexed}	
	|	<RGB> {if COLOUR SELECTION MODE is direct}	
<integer: colour-index>	=	<non-negative integer>	

NOTE - Edge types with values above 5 are reserved for registration.

Page 40

Add the following after 8.4.6:

8.4.7 LINE CLIPPING MODE

<LINE-CLIPPING-MODE-opcode: 3/3 2/6>
<enumerated: clipping-mode>

<enumerated: clipping-mode>	=	<integer:0>	{locus}
	|	<integer:1>	{shape}
	|	<integer:2>	{locus then shape}

8.4.8 MARKER CLIPPING MODE

<MARKER-CLIPPING-MODE-opcode: 3/3 2/7>
<enumerated: clipping-mode>

<enumerated: clipping-mode>	=	<integer:0>	{locus}
	|	<integer:1>	{shape}
	|	<integer:2>	{locus then shape}

8.4.9 EDGE CLIPPING MODE

<EDGE-CLIPPING-MODE-opcode: 3/3 2/8>
<enumerated: clipping mode>

<enumerated: clipping mode>	=	<integer:0>	{locus}
	|	<integer:1>	{shape}
	|	<integer:2>	{locus then shape}

8.4.10 NEW REGION

<NEW-REGION-opcode: 3/3 2/9>

8.4.11 SAVE PRIMITIVE CONTEXT

<SAVE-PRIMITIVE-CONTEXT-opcode: 3/3 2/10>
<name: context>

<name: context> = <integer>

8.4.12 RESTORE PRIMITIVE CONTEXT

<RESTORE-PRIMITIVE-CONTEXT-opcode: 3/3 2/11>
<name: context>

<name: context> = <integer>

Page 45

Add the following after 8.5.19:

8.5.20 CIRCULAR ARC CENTRE REVERSED

<CIRCULAR-ARC-CENTRE-REVERSED-opcode: 3/4 2/8>
<point: centrepoint>
<VDC: DX_start>
<VDC: DY_start>
<VDC: DX_end>
<VDC: DY_end>
<VDC: radius>

8.5.21 CONNECTING EDGE

<CONNECTING-EDGE-opcode: 3/4 2/9>

Page 46

Sub-clause 8.6.2: Add the following note at the end:

NOTE - Line types with values above 5 are reserved for registration.

Page 47

Sub-clause 8.6.6: Add the following note at the end:

NOTE - Marker types with values above 5 are reserved for registration.

Page 50

Sub-clause 8.6.24: Add the following note at the end:

NOTE - Hatch indices with values above 6 are reserved for registration.

Page 50

Sub-clause 8.6.27: Add the following note at the end:

NOTE - Edge types with values above 5 are reserved for registration.

Page 54

Add the following after 8.6.35:

8.6.36 PICK IDENTIFIER

<PICK-ID-opcode: 3/6 3/2>
<name: pick-identifier>

<name: pick-identifier>	=	<integer>	

Page 55

Add the following after 8.8:

8.9 Segment elements

8.9.1 COPY SEGMENT

<COPY-SEGMENT-opcode: 3/8 2/0>
<name: segment-identifier>
<transformation-matrix>
<enumerated: segment-transformation-application>

<name: segment-identifier>	=	<integer>	
<transformation-matrix>	=	<real: a11 >	
		<real: a12 >	
		<real: a21 >	
		<real: a22 >	
		<vdc : a13 >	
		<vdc : a23 >	
<enumerated:segment-transformation-application>	=	<integer:0>	{no}
	|	<integer:1>	{yes}

8.9.2 INHERITANCE FILTER

<INHERITANCE-FILTER-opcode: 3/8 2/1>
<enumerated: filter-selection-list>+
<enumerated: selection-setting>

<enumerated: filter-selection-list>	=	<integer:0>	{line bundle index}
	|	<integer:1>	{line type}
	|	<integer:2>	{line width}
	|	<integer:3>	{line colour}
	|	<integer:4>	{line clipping mode}
	|	<integer:5>	{marker bundle index}
	|	<integer:6>	{marker type}
	|	<integer:7>	{marker size}
	|	<integer:8>	{marker colour}
	|	<integer:9>	{marker clipping mode}
	|	<integer:10>	{text bundle index}
	|	<integer:11>	{text font index}
	|	<integer:12>	{text precision}
	|	<integer:13>	{character expansion factor}
	|	<integer:14>	{character spacing}
	|	<integer:15>	{text colour}
	|	<integer:16>	{character height}

	|	<integer:17>	{character orientation}
	|	<integer:18>	{text path}
	|	<integer:19>	{text alignment}
	|	<integer:20>	{fill bundle index}
	|	<integer:21>	{interior style}
	|	<integer:22>	{fill colour}
	|	<integer:23>	{hatch index}
	|	<integer:24>	{pattern index}
	|	<integer:25>	{edge bundle index}
	|	<integer:26>	{edge type}
	|	<integer:27>	{edge width}
	|	<integer:28>	{edge colour}
	|	<integer:29>	{edge visibility}
	|	<integer:30>	{edge clipping mode}
	|	<integer:31>	{fill reference point}
	|	<integer:32>	{pattern size}
	|	<integer:33>	{auxiliary colour}
	|	<integer:34>	{transparency}
	|	<integer:35>	{line attributes}
	|	<integer:36>	{marker attributes}
	|	<integer:37>	{text presentation and placement attributes}
	|	<integer:38>	{text placement and orientation attributes}
	|	<integer:39>	{fill attributes}
	|	<integer:40>	{edge attributes}
	|	<integer:41>	{pattern attributes}
	|	<integer:42>	{output control}
	|	<integer:43>	{pick identifier}
	|	<integer:44>	{all attributes and control}
	|	<integer:45>	{all}
	|	<integer:46>	{line type ASF}
	|	<integer:47>	{line width ASF}
	|	<integer:48>	{line colour ASF}
	|	<integer:49>	{marker type ASF}
	|	<integer:50>	{marker size ASF}
	|	<integer:51>	{marker colour ASF}
	|	<integer:52>	{text font index ASF}
	|	<integer:53>	{text precision ASF}
	|	<integer:54>	{character expansion factor ASF}
	|	<integer:55>	{character spacing ASF}
	|	<integer:56>	{text colour ASF}
	|	<integer:57>	{interior style ASF}
	|	<integer:58>	{fill colour ASF}
	|	<integer:59>	{hatch index ASF}
	|	<integer:60>	{pattern index ASF}
	|	<integer:61>	{edge type ASF}
	|	<integer:62>	{edge width ASF}
	|	<integer:63>	{edge colour ASF}
	|	<integer:64>	{line ASFs}
	|	<integer:65>	{marker ASFs}
	|	<integer:66>	{text ASFs}
	|	<integer:67>	{fill ASFs}
	|	<integer:68>	{edge ASFs}
	|	<integer:69>	{all ASFs}
<enumerated: selection-setting>	=	<integer:0>	{state list}
	|	<integer:1>	{segment}

8.9.3 CLIP INHERITANCE

```
<CLIP-INHERITANCE-opcode: 3/8 2/2>
<enumerated: clip-inheritance>
```

<enumerated: clip inheritance>	=	<integer:0>	{state list}
	|	<integer:1>	{intersection}

8.9.4 SEGMENT TRANSFORMATION

```
<SEGMENT-TRANSFORMATION-opcode: 3/8 2/3>
<name: segment-identifier>
<transformation-matrix>
```

<name: segment-identifier>	=	<integer>
<transformation-matrix>	=	<real: a11 >
		<real: a12 >
		<real: a21 >
		<real: a22 >
		<vdc : a13 >
		<vdc : a23 >

8.9.5 SEGMENT HIGHLIGHTING

```
<SEGMENT-HIGHLIGHTING-opcode: 3/8 2/4>
<name: segment-identifier>
<enumerated: segment-highlighting>
```

<name:segment-identifier>	=	<integer>	
<enumerated: segment-highlighting>	=	<integer: 0>	{normal}
	|	<integer: 1>	{highlighted}

8.9.6 SEGMENT DISPLAY PRIORITY

```
<SEGMENT-DISPLAY-PRIORITY-opcode: 3/8 2/5>
<name: segment-identifier>
<integer: segment-display-priority>
```

<name:segment-identifier>	=	<integer>
<integer: segment-display-priority>	=	<positive integer>

8.9.7 SEGMENT PICK PRIORITY

```
<SEGMENT-PICK-PRIORITY-opcode: 3/8 2/6>
<name: segment-identifier>
<integer: pick-priority>
```

<name:segment-identifier>	=	<integer>
<integer: pick-priority>	=	<positive integer>

Page 56

Add the following at the end of clause 9

NAME PRECISION : 10

Page 60

Add the following at the end of Annex A

<name precision value>	::=	<integer> {see 8.2}
<viewport point>	::=	<integer><integer>
	|	<real><real>
<VC value>	::=	<integer>
	|	<real>
<name>	::=	<integer>
<2 x 2 matrix of reals>	::=	<real>(4) {see 8.9}
<2 x 1 matrix of vdcs>	::=	<vdc value>(2) {see 8.9}

ISO/IEC 8632-3 : 1987/Am.1 : 1990

Information processing systems - Computer graphics - Metafile for the storage and transfer of picture descriptive information -

Part 3:
Binary Encoding

Amendment 1

Page 16

Add the following at the end of table 1:

Abstract symbol	Parameter construction from	Octets per parameter: symbol and value	Parameter range: symbol and value
N	SI at integer precision (np)	BN {=np/8}	NR {-2**(np-1) to 2**(np-1)-1}
VC	I or R	BVC {=BI} or BVC {=BR}	VCR {=IR} {see note 13} or VCR {=RR}
VP	(VC,VC)	BVP {=2*BVC}	VCR {see notes 1,13,14}

Page 16

Add the following to the additional description ("notes") for table 1:

13 The abstract parameter type VC, a single VC value, is either a real or an integer, depending on the declaration of the picture descriptor element DEVICE VIEWPORT SPECIFICATION MODE. When DEVICE VIEWPORT SPECIFICATION MODE is 'fraction of display surface', the value is real. When DEVICE VIEWPORT SPECIFICATION MODE is 'millimetres with scale factor' or 'physical device coordinates', the value is integer. Subsequent tables use a single set of values, VC, BVC and VCR, recognising that they are computed differently depending on DEVICE VIEWPORT SPECIFICATION MODE.

14 The abstract parameter type VC is a single value; a viewport point, VP, is an ordered pair of VC.

Page 19

Add the following at the end of table 2:

8 Segment Control and Segment Attribute elements

Page 20

Add the following at the end of table 3:

Element class 0	Element Id	Parameter type	Parameter list length	Parameter range	Default
BEGIN SEGMENT	6	N	BN	NR	n/a
END SEGMENT	7	n/a	0	n/a	n/a
BEGIN FIGURE	8	n/a	0	n/a	n/a
END FIGURE	9	n/a	0	n/a	n/a

Code Description

6 BEGIN SEGMENT: has 1 parameter:

P1: (name) segment identifier

7 END SEGMENT: has no parameters

8 BEGIN FIGURE: has no parameters

9 END FIGURE: has no parameters

Page 21

Add the following at the end of table 4:

Element class 1	Element Id	Parameter type	Parameter list length	Parameter range	Default
NAME PRECISION	16	N	BN	8,16,24,32	16
MAXIMUM VDC EXTENT EXTENT	17	2P	2BP	VDCR	VDC
SEGMENT PRIORITY EXTENT	18	2I	2BI	IR	0, 255

Code Description

16 NAME PRECISION: has 1 parameter:

P1: (name) name precision: 8, 16, 24 or 32 are the only valid values

17 MAXIMUM VDC EXTENT: has 2 parameters:

P1: (point) first point
P2: (point) second point

18 SEGMENT PRIORITY EXTENT: has 2 parameters:

P1: (integer) minimum segment priority value
P2: (integer) maximum segment priority value

Page 22

Add to the note P2 of METAFILE ELEMENT LIST:

version-2 set	(-1,2)
extended-primitives set	(-1,3)
version-2-gksm set	(-1,4)

Page 24

Add to the end of the note P2 for SCALING MODE:

NOTE - This parameter is always encoded as Floating Point, regardless of the value of the fixed/floating flag of REAL PRECISION. If a REAL PRECISION (floating, n, m) has preceded, then the precision used is n,m. If a REAL PRECISION element for floating point has not preceded, then the default precision is used.

Page 24

Add the following at the end of table 5:

Element class 2	Element Id	Parameter type	Parameter list length	Parameter range	Default
DEVICE VIEWPORT	8	2VP	2BVP	VCR	see below
DEVICE VIEWPORT SPECIFICATION MODE	9	E,R(FP)	BE+BFP	{0,1,2},FPR	0,-
DEVICE VIEWPORT MAPPING	10	3E	3BE	{0,1} {0,1,2} {0,1,2}	1 0 0
LINE REPRESENTATION	11	2IX, (VDC or (BVDC or R),CO	2BIX+ ++VDCR or BR)+BCO	+IXR,IXR, ++RR,COR	n/a
MARKER REPRESENTATION	12	2IX, (VDC or (BVDC or R),CO	2BIX+ ++VDCR or BR)+BCO	+IXR,IXR, ++RR,COR	n/a
TEXT REPRESENTATION	13	2IX, E, 2R,CO 2BR+BCO	2BIX+ BE+	+IXR, {0,1,2}, +RR,RR,COR	n/a
FILL REPRESENTATION	14	IX, E,CO, 2IX	BIX+ BE+BCO+ 2BIX	+IXR, {0..4},COR, IXR,+IXR	n/a
EDGE REPRESENTATION	15	2IX, (VDC or (BVDC or R),CO	2BIX+ ++VDCR or BR)+BCO	+IXR,IXR, ++RR,COR	n/a

Code Description

8 DEVICE VIEWPORT: has 2 parameters:

P1: (viewport point) first point
P2: (viewport point) second point

9 DEVICE VIEWPORT SPECIFICATION MODE: has 2 parameters:

P1: (enumerated) VC specifier: valid values are:
- 0 fraction of drawing surface
- 1 millimetres with scale factor
- 2 physical device coordinates

P2: (real) metric scale factor, ignored if P1=0 or P1=2

NOTE - This parameter is always encoded as Floating Point, regardless of the value of the fixed/floating flag of REAL PRECISION. If a REAL PRECISION (floating, n, m) has preceded, then the precision used is n,m. If a REAL PRECISION element for floating point has not preceded, then the default precision is used.

10 DEVICE VIEWPORT MAPPING: has 3 parameters:

P1: (enumerated) isotropy flag: valid values are:
- 0 not forced
- 1 forced

P2: (enumerated) horizontal alignment flag: valid values are:
- 0 left
- 1 centre
- 2 right

P3: (enumerated) vertical alignment flag: valid values are:
0 bottom
1 centre
2 top

11 LINE REPRESENTATION: has 4 parameters:

P1: (index) line bundle index
P2: (index) line type: the following values are standardized:
1 solid
2 dash
3 dot
4 dash-dot
5 dash-dot-dot
negative for private use
P3: (vdc or real) absolute line width or line width scale factor
P4: (colour) line colour: its form depends on COLOUR SELECTION MODE.

NOTE - Line types with values above 5 are reserved for registration.

12 MARKER REPRESENTATION: has 4 parameters:

P1: (index) marker bundle index
P2: (index) marker type: the following values are standardized:
1 dot
2 plus
3 asterisk
4 circle
5 cross
negative for private use
P3: (vdc or real) absolute marker width or marker size scale factor
P4: (colour) marker colour: its form depends on COLOUR SELECTION MODE.

NOTE - Marker types with values above 5 are reserved for registration.

13 TEXT REPRESENTATION: has 6 parameters:

P1: (index) text bundle index
P2: (index) text font index
P3: (enumerated) text precision: valid values are:
0 string
1 character
2 stroke
P4: (real) character spacing
P5: (real) character expansion factor
P6: (colour) text colour; its form depends on COLOUR SELECTION MODE

14 FILL REPRESENTATION: has 5 parameters:

P1: (index) fill area bundle index
P2: (enumerated) interior style: valid values are:
0 hollow
1 solid
2 pattern
3 hatch
4 empty
P3: (colour) fill colour: its form depends on COLOUR SELECTION MODE
P4: (index) hatch index: the following values are standardized:
1 horizontal
2 vertical
3 positive slope
4 negative slope

5 combined vertical and horizontal slant
6 combined left and right slant
negative for private use
P5: (index) pattern index

NOTE - Hatch indices with values above 6 are reserved for registration.

15 EDGE REPRESENTATION: has 4 parameters:

P1: (index) edge bundle index
P2: (index) edge type: the following values are standardized:
1 solid
2 dash
3 dot
4 dash-dot
5 dash-dot-dot
negative for private use
P3: (vdc or real) absolute edge width or line width scale factor
P4: (colour) edge colour: its form depends on COLOUR SELECTION MODE.

NOTE - Edge types with values above 5 are reserved for registration.

Page 26

Add the following at the end of table 6:

Element class 3	Element Id	Parameter type	Parameter list length	Parameter range	Default
LINE CLIPPING MODE	7	E	BE	{0,1,2}	0
MARKER CLIPPING MODE	8	E	BE	{0,1,2}	0
EDGE CLIPPING MODE	9	E	BE	{0,1,2}	0
NEW REGION	10	n/a	0	n/a	n/a
SAVE PRIMITIVE CONTEXT	11	N	BN	NR	n/a
RESTORE PRIMITIVE CONTEXT	12	N	BN	NR	n/a

7 LINE CLIPPING MODE: has 1 parameter:

P1: (enumerated) clipping mode: valid values are:
0 locus
1 shape
2 locus then shape

8 MARKER CLIPPING MODE: has 1 parameter:

P1: (enumerated) clipping mode: valid values are:
0 locus
1 shape
2 locus then shape

9 EDGE CLIPPING MODE: has 1 parameter:

P1: (enumerated) clipping mode: valid values are:
0 locus
1 shape
2 locus then shape

10 NEW REGION: has no parameters

11 SAVE PRIMITIVE CONTEXT: has 1 parameter:

P1: (name) context name

12 RESTORE PRIMITIVE CONTEXT: has 1 parameter:

P1: (name) context name

Page 28

Add the following at the end of table 7:

Element class 4	Element Id	Parameter type	Parameter list length	Parameter range	Default
CIRCULAR ARC CENTRE REVERSED	20	P,4VDC, VDC	BP+4BVDC+ BVDC	VDCR,VDCR, ++VDCR	n/a
CONNECTING EDGE	21	n/a	0	n/a	n/a

Code Description

20 CIRCULAR ARC CENTRE REVERSED: has 6 parameters:
P1: (point) centre of circle
P2: (vdc) delta X for start vector
P3: (vdc) delta Y for start vector
P4: (vdc) delta X for end vector
P5: (vdc) delta Y for end vector
P6: (vdc) radius of circle

21 CONNECTING EDGE: has no parameters

Page 33

Add the following at the end of table 8:

Element class 5	Element Id	Parameter type	Parameter list length	Parameter range	Default
PICK IDENTIFIER	36	N	BN	NR	0

Code Description

36 PICK IDENTIFIER: has 1 parameter:
P1: (name) pick identifier

Page 33

Sub-clause 7.7: Add the following note after code 1 LINE TYPE:

NOTE - Line types with values above 5 are reserved for registration.

Page 34

Sub-clause 7.7: Add the following note after code 6 MARKER TYPE:

NOTE - Marker types with values above 5 are reserved for registration.

Page 35

Sub-clause 7.7: Add the following note after code 24 HATCH INDEX:

NOTE - Hatch indices with values above 6 are reserved for registration.

Page 36

Sub-clause 7.7: Add the following note after code 27 EDGE TYPE:

NOTE - Edge types with values above 5 are reserved for registration.

Page 39

Add the following after 7.9:

7.10 Segment control and segment attribute elements

Table 11 - Encoding of segment control and segment attribute elements

Element class 8	Element Id	Parameter type	Parameter list length	Parameter range	Default
COPY SEGMENT	1	N,4R, 2VDC, E	BN+4BR+ 2BVDC + BE	NR,RR, VDCR, {0,1}	-,0
INHERITANCE FILTER	2	nE,E	(n+1)BE	{0..69},{0,1}	-,1
CLIP INHERITANCE	3	E	BE	{0,1}	0
SEGMENT TRANSFORMATION	4	N,4R, 2VDC	BN+4BR+ 2BVDC	NR,RR, VDCR	n/a,1,0,0,1 0,0
SEGMENT HIGHLIGHTING	5	N,E	BN+BE	NR,{0,1}	n/a,0
SEGMENT DISPLAY PRIORITY	6	N,I	BN+BI	NR,IR	n/a, see below
SEGMENT PICK PRIORITY	7	N,I	BN+BI	NR,IR	n/a, see below

Code Description

1 COPY SEGMENT: has 3 parameters:

P1: (name) segment identifier
P2: The next 6 values are components of a transofrmation matrix consisting of a scaling and rotation portion (2 x 2 R) and a translation portion (2 x 1 VDC). In the binary encoding this is expressed as a 2 x 3 matrix of the form:

a11: (real) x scale component
a12: (real) x rotation component
a21: (real) y rotation component
a22: (real) y scale component
a13: (vdc) x translation component
a23: (vdc) y translation component

P3: (enumerated) segment transformation application: valid values are:
0: no
1: yes

2 INHERITANCE FILTER: has two parameters. The first is a list of up to 70 attribute or group designators. The second is a single setting value.
P1: (enumerated list) list of one or more of:
0 line bundle index
1 line type
2 line width
3 line colour
4 line clipping mode
5 marker bundle index
6 marker type

7	marker size
8	marker colour
9	marker clipping mode
10	text bundle index
11	text font index
12	text precision
13	character expansion factor
14	character spacing
15	text colour
16	character height
17	character orientation
18	text path
19	text alignment
20	fill bundle index
21	interior style
22	fill colour
23	hatch index
24	pattern index
25	edge bundle index
26	edge type
27	edge width
28	edge colour
29	edge visibility
30	edge clipping mode
31	fill reference point
32	pattern size
33	auxiliary colour
34	transparency
35	line attributes
36	marker attributes
37	text presentation and placement attributes
38	text placement and orientation attributes
39	fill attributes
40	edge attributes
41	pattern attributes
42	output control
43	pick identifier
44	all attributes and control
45	all
46	line type asf
47	line width asf
48	line colour asf
49	marker type asf
50	marker size asf
51	marker colour asf
52	text font index asf
53	text precision asf
54	character expansion factor asf
55	character spacing asf
56	text colour asf
57	interior style asf
58	fill colour asf
59	hatch index asf
60	pattern index asf
61	edge type asf
62	edge width asf
63	edge colour asf
64	line asfs
65	marker asfs
66	text asfs
67	fill asfs

68 edge asfs
69 all asfs

P2: (enumerated) setting: valid values are:
0 state list
1 segment

3 CLIP INHERITANCE: has 1 parameter
P1: (enumerated) clip inheritance: valid values are:
0 state list
1 intersection

4 SEGMENT TRANSFORMATION: has 2 parameters:

P1: (name) segment identifier
P2: The next 6 values are components of a transofrmation matrix consisting of a scaling and rotation portion (2 x 2 R) and a translation portion (2 x 1 VDC). In the binary encoding this is expressed as a 2 x 3 matrix of the form:

a11: (real) x scale component
a12: (real) x rotation component
a21: (real) y rotation component
a22: (real) y scale component
a13: (vdc) x translation component
a23: (vdc) y translation component

5 SEGMENT HIGHLIGHTING: has 2 parameters:

P1: (name) segment identifier
P2: (enumerated) highlighting: valid values are:
0 normal
1 highlighted

6 SEGMENT DISPLAY PRIORITY: has 2 parameters:

P1: (name) segment identifier
P2: (integer) segment display priority
The default of the segment display priority is equal to the minimum segment priority value (see 7.3)

7 SEGMENT PICK PRIORITY: has 2 parameters:

P1: (name) segment identifier
P2: (integer) segment pick priority
The default of the segment pick priority is equal to the minimum segment priority value (see 7.3)

Page 40

Add the following at the end of clause 8:

NAME PRECISION 16 bits

Page 43

Add the following at the end of the list of table references:

<viewport point> ::= <integer(2)>|<real(2)>

<vc value> ::= <integer>|<real>

<name>	::=	<integer>
<2x2 matrix of reals>	::=	<real>(4)
<2x1 matrix of vdcs>	::=	<vdc value>(2)

Page 48

Add the following to the list of elements:

Class	Element Code	Element Name
0	6	BEGIN SEGMENT
0	7	END SEGMENT
0	8	BEGIN FIGURE
0	9	END FIGURE
1	16	NAME PRECISION
1	17	MAXIMUM VDC EXTENT
1	18	SEGMENT PRIORITY EXTENT
2	8	DEVICE VIEWPORT
2	9	DEVICE VIEWPORT SPECIFICATION MODE
2	10	DEVICE VIEWPORT MAPPING
2	11	LINE REPRESENTATION
2	12	MARKER REPRESENTATION
2	13	TEXT REPRESENTATION
2	14	FILL REPRESENTATION
2	15	EDGE REPRESENTATION
3	7	LINE CLIPPING MODE
3	8	MARKER CLIPPING MODE
3	9	EDGE CLIPPING MODE
3	10	NEW REGION
3	11	SAVE PRIMITIVE CONTEXT
3	12	RESTORE PRIMITIVE CONTEXT
4	20	CIRCULAR ARC CENTRE REVERSED
4	21	CONNECTING EDGE
5	36	PICK IDENTIFIER
8	1	COPY SEGMENT
8	2	INHERITANCE FILTER
8	3	CLIP INHERITANCE
8	4	SEGMENT TRANSFORMATION
8	5	SEGMENT HIGHLIGHTING
8	6	SEGMENT DISPLAY PRIORITY
8	7	SEGMENT PICK PRIORITY

ISO/IEC 8632-4 : 1987/Am.1 : 1990

Information processing systems - Computer graphics - Metafile for the storage and transfer of picture descriptive information -

Part 4: Clear Text Encoding

Amendment 1

Page 8

Sub-clause 5.3.1. Change the text in the first sentence from : "INTEGERS, INTEGER COORDINATES, INDICES, and ...." to:

INTEGERS, INTEGER COORDINATES, INDICES, NAMES, and.....

Page 11

Add the following to the end of 5.3.5

N ::= <I> {name}

VC ::= <R>|<I> {viewport coordinate data}

{The abstract parameter type VC, a single VC value, is either a real or an integer, depending on the declaration of the picture descriptor element DEVICE VIEWPORT SPECIFICATION MODE. When DEVICE VIEWPORT SPECIFICATION MODE is 'fraction of display surface', the value is real. When DEVICE VIEWPORT SPECIFICATION MODE is 'millimetres with scale factor' or 'physical device coordinates', the value is integer.}

VPOINTREC ::= <VC><SEP><VC>

VP ::= <VPOINTREC>|< <LEFT PAREN><OPTSEP><VPOINTREC><OPTSEP>
<RIGHT PAREN> >

{COORDINATE in viewport coordinate space. Parentheses are optional. If they are used, they shall group exactly two real or integer numbers, depending on DEVICE VIEWPORT SPECIFICATION MODE. The parenthesized form is intended to aid readability of the metafile.}

TM ::= <<R: a11>
<SEP>
<R: a12>
<SEP>
<R: a21>
<SEP>
<R: a22>
<SEP>
<VDC: a13>
<SEP>
<VDC: a23>>

Page 12

Add the following at the end of 5.4.3

ALL
COPY
FIGURE

FILTER
FORCED
FRACTION
GKSM
INTERSECTION
LOCUS
MATRIX
NAME
NEW
OUTPUT
PICK
REGION
SAVE
SHAPE
THEN

Page 12

Add the following at the end of 5.4.4:

ATTRIBUTE(S)	ATTR
CLIPPING	CLIP
CONNECTING	CONN
CONTEXT	CONT
COORDINATE(S)	COORD
DEVICE	DEV
DISPLAY	DISP
EXTENDED	EXT
HIGHLIGHTING	HIGHL
IDENTIFIER	ID
INHERITANCE	INH
MAPPING	MAP
MILLIMETRE	MM
PHYSICAL	PHY
PLACEMENT	PLACEM
PRESENTATION	PRES
PRIMITIVE(S)	PRIM
PRIORITY	PRI
REPRESENTATION	REP
RESTORE	RES
REVERSED	REV
SEGMENT	SEG
STATELIST	STLIST
TRANSFORMATION	TRAN
TWO	2
VIEWPORT	VP

Page 14

Add the following at the end of 5.4.5:

BEGIN SEGMENT	BEGSEG
END SEGMENT	ENDSEG
BEGIN FIGURE	BEGFIGURE
END FIGURE	ENDFIGURE
NAME PRECISION	NAMEPREC
MAXIMUM VDC EXTENT	MAXVDCEXT
SEGMENT PRIORITY EXTENT	SEGPRIEXT
DEVICE VIEWPORT	DEVVP
DEVICE VIEWPORT SPECIFICATION MODE	DEVVPMODE
DEVICE VIEWPORT MAPPING	DEVVPMAP
LINE REPRESENTATION	LINEREP

MARKER REPRESENTATION	MARKERREP
TEXT REPRESENTATION	TEXTREP
FILL REPRESENTATION	FILLREP
EDGE REPRESENTATION	EDGEREP
LINE CLIPPING MODE	LINECLIPMODE
MARKER CLIPPING MODE	MARKERCLIPMODE
EDGE CLIPPING MODE	EDGECLIPMODE
NEW REGION	NEWREGION
SAVE PRIMITIVE CONTEXT	SAVEPRIMCONT
RESTORE PRIMITIVE CONTEXT	RESPRIMCONT
CIRCULAR ARC CENTRE REVERSED	ARCCTRREV
CONNECTING EDGE	CONNEDGE
PICK IDENTIFIER	PICKID
COPY SEGMENT	COPYSEG
INHERITANCE FILTER	INHFILTER
CLIP INHERITANCE	CLIPINH
SEGMENT TRANSFORMATION	SEGTRAN
SEGMENT VISIBILITY	SEGVIS
SEGMENT HIGHLIGHTING	SEGHIGHL
SEGMENT DISPLAY PRIORITY	SEGDISPPRI
SEGMENT PICK PRIORITY	SEGPICKPRI

Page 15

Add the following at the end of 6.2:

```
BEGIN SEGMENT          ::=  BEGSEG
                                 <SOFTSEP>
                                 <N:SEGID>
                            <TERM>

END SEGMENT            ::=  ENDSEG  <TERM>

BEGIN FIGURE           ::=  BEGFIGURE <TERM>

END FIGURE             ::=  ENDFIGURE <TERM>
```

Page 17

Add at the end of METAFILE ELEMENT LIST:

The words VERSION2, EXTPRIM and VERSION2GKSM may also be used in this string

Page 17

Add the following at the end of sub-clause 6.3:

```
NAME PRECISION         ::=  NAMEPREC
                                 <SOFTSEP>
                                 <I:MININT>
                                 <SEP>
                                 <I:MAXINT>
                            <TERM>
```

```
MAX VDC EXTENT              ::= MAXVDCEXT
                                    <SOFTSEP>
                                    <P:FIRSTCORNER>
                                    <SEP>
                                    <P:SECONDCORNER>
                                <TERM>

SEGMENT PRIORITY EXTENT     ::= SEGPRIEXT
                                    <SOFTSEP>
                                    <I:MINSEGPRI>
                                    <SEP>
                                    <I:MAXSEGPRI>
                                <TERM>
```

Page 18

Add the following at the end of 6.4

```
DEVICE VIEWPORT             ::= DEVVP
                                    <SOFTSEP>
                                    <VP:FIRSTCORNER>
                                    <SEP>
                                    <VP:SECONDCORNER>
                                <TERM>

DEVICE VIEWPORT SPECIFICATION
MODE                        ::= DEVVPMODE
                                    <SOFTSEP>
                                    <FRACTION|MM|PHYDEVCOORD>
                                    <SEP>
                                    <R:SCALEFACTOR>
                                <TERM>

DEVICE VIEWPORT MAPPING     ::= DEVVPMAP
                                    <SOFTSEP>
                                    <NOTFORCED|FORCED>
                                    <SEP>
                                    <LEFT|CTR|RIGHT>
                                    <SEP>
                                    <BOTTOM|CTR|TOP>
                                <TERM>

LINE REPRESENTATION         ::= LINEREP
                                    <SOFTSEP>
                                    <I:BUNDLEINDEX> {positive}
                                    <SEP>
                                    <I:LINETYPE>
                                          {1=solid, 2=dash
                                          3=dot, 4=dash-dot
                                          5=dash-dot-dot
                                          <0 implementation dependent}
                                    <SEP>
                                    <V:LINEWIDTH> {non-negative}
                                    <SEP>
                                    <K:LINECOLR>
                                <TERM>
```

NOTE - Line types with values above 5 are reserved for registration.

```
MARKER REPRESENTATION       ::= MARKERREP
                                    <SOFTSEP>
```

```
                  <I:BUNDLEINDEX> {positive}
                  <SEP>
                  <I:MARKERTYPE>
                          {1=dot, 2=plus
                          3=asterisk, 4=circle
                          5=cross (x)
                          <0 implementation dependent}
                  <SEP>
                  <V:MARKERSIZE> {non-negative}
                  <SEP>
                  <K:MARKERCOLR>
              <TERM>
```

NOTE - Marker types with values above 5 are reserved for registration.

```
TEXT REPRESENTATION     ::= TEXTREP
                  <SOFTSEP>
                  <I:BUNDLEINDEX> {positive}
                  <SEP>
                  <I:FONTINDEX> {positive}
                  <SEP>
                  <STRING|CHAR|STROKE>
                  <SEP>
                  <R:SPACING>
                  <SEP>
                  <R:FACTOR>
                  <SEP>
                  <K:TEXTCOLR>
              <TERM>

FILL REPRESENTATION     ::= FILLREP
                  <SOFTSEP>
                  <I:BUNDLEINDEX> {positive}
                  <SEP>
                  <HOLLOW|SOLID|PAT|HATCH|EMPTY>
                  <SEP>
                  <K:FILLCOLR>
                  <SEP>
                  <I:HATCHINDEX>
                          {1=horizontal,2=vertical
                          3=positive slope
                          4=negative slope
                          5=horizontal/vertical cross
                          6=+/- slope cross
                          <0 implementation dependent
                  <SEP>
                  <I:PATINDEX> {positive}
              <TERM>
```

NOTE - Hatch indices with values above 6 are reserved for registration.

```
EDGE REPRESENTATION     ::= EDGEREP
                  <SOFTSEP>
                  <I:BUNDLEINDEX> {positive}
                  <SEP>
                  <I:EDGETYPE>
                          {1=solid, 2=dash
                          3=dot, 4=dash-dot
                          5=dash-dot-dot
                          <0 implementation dependent}
                  <SEP>
                  <V:EDGEWIDTH> {non-negative}
```

```
                                <SEP>
                                <K:EDGECOLR>
                            <TERM>
```

NOTE - Edge types with values above 5 are reserved for registration.

Page 19

Add the following at the end of 6.5

```
LINE CLIPPING MODE          ::= LINECLIPMODE
                                <SOFTSEP>
                                <LOCUS|SHAPE|LOCUSTHENSHAPE>
                            <TERM>

MARKER CLIPPING MODE        ::= MARKERCLIPMODE
                                <SOFTSEP>
                                <LOCUS|SHAPE|LOCUSTHENSHAPE>
                            <TERM>

EDGE CLIPPING MODE          ::= EDGECLIPMODE
                                <SOFTSEP>
                                <LOCUS|SHAPE|LOCUSTHENSHAPE>
                            <TERM>

NEW REGION                  ::= NEWREGION <TERM>

SAVE PRIMITIVE CONTEXT      ::= SAVEPRIMCONT
                                <SOFTSEP>
                                <I:CONTEXTNAME>
                            <TERM>

RESTORE PRIMITIVE
  CONTEXT                   ::= RESPRIMCONT
                                <SOFTSEP>
                                <I:CONTEXTNAME>
                            <TERM>
```

Page 24

Add the following at the end of 6.6

```
CIRCULAR ARC CENTRE
  REVERSED                  ::= ARCCTRREV
                                <CTRARCSPEC>
                            <TERM>

CONNECTING EDGE             ::= CONNEDGE  <TERM>
```

Page 24

Sub-clause 6.7: Add the following note after the description of LINE TYPE:

NOTE - Line types with values above 5 are reserved for registration.

Page 24

Sub-clause 6.7: Add the following note after the description of MARKER TYPE:

NOTE - Marker types with values above 5 are reserved for registration.

Page 26

Sub-clause 6.7: Add the following note after the description of HATCH INDEX:

NOTE - Hatch indices with values above 6 are reserved for registration.

Page 27

Sub-clause 6.7: Add the following note after the description of EDGE TYPE:

NOTE - Edge types with values above 5 are reserved for registration.

Page 28

Add the following at the end of 6.7

```
PICK IDENTIFIER          ::= PICKID
                                <SOFTSEP>
                                <I:PICKID>
                             <TERM>
```

Page 29

Add the following after 6.9:

6.10 Encoding segment control and segment attribute elements

```
COPY SEGMENT             ::= COPYSEG
                                <SOFTSEP>
                                <I:SEGID>
                                <SEP>
                                <TM:TRANMATRIX>
                                <NO|YES>
                             <TERM>
```

```
INHERITANCE FILTER       ::= INHFILTER
                                     <SOFTSEP>
                                     <ELEMORGROUPNAME>
                                     <<SEP><ELEMORGROUPNAME>>*
                                     <SEP>
                                     <STLIST|SEG>
                             <TERM>
```

```
ELEMORGROUPNAME          ::= <LINEINDEX|
                             LINETYPE|
                             LINEWIDTH|
                             LINECOLR|
                             LINECLIPMODE|
                             MARKERINDEX|
                             MARKERTYPE|
                             MARKERSIZE|
                             MARKERCOLR|
                             MARKERCLIPMODE|
                             TEXTINDEX|
                             TEXTFONTINDEX|
                             TEXTPREC|
                             CHARACTEREXPAN|
                             CHARACTERSPACE|
                             TEXTCOLR|
```

```
CHARHEIGHT|
CHARORI|
TEXTPATH|
TEXTALIGN|
FILLINDEX|
INTSTYLE|
FILLCOLR|
HATCHINDEX|
PATINDEX|
EDGEINDEX|
EDGETYPE|
EDGEWIDTH|
EDGECOLR|
EDGEVIS|
EDGECLIPMODE|
FILLREFPT|
PATSIZE|
AUXCOLR|
TRANSPARENCY|
LINEATTR|
MARKERATTR|
TEXPRESANDPLACEMATTR|
TEXTPLACEMANDORIATTR|
FILLATTR|
EDGEATTR|
PATATTR|
OUTPUTCTRL|
PICKID|
ALLATTRCTRL|
ALLINH|
LINETYPEASF|
LINEWIDTHASF|
LINECOLRASF|
MARKERTYPEASF|
MARKERSIZEASF|
MARKERCOLRASF|
TEXTFONTINDEXASF|
TEXTPRECASF|
CHARACTEREXPANASF|
CHARACTERSPACEASF|
TEXTCOLRASF|
INTSTYLEASF|
FILLCOLRASF|
HATCHINDEXASF|
PATINDEXASF|
EDGETYPEASF|
EDGEWIDTHASF|
EDGECOLRASF|
ALLLINE|
ALLMARKER|
ALLTEXT|
ALLFILL|
ALLEDGE|
ALL>
```

NOTE - ALLINH means all attributes, control elements and ASFs. ALLLINE, ALLMARKER, ALLTEXT, ALLFILL, ALLEDGE and ALL have the meaning defined in 6.7.

```
CLIP INHERITANCE        ::= <CLIPINH
                                <SOFTSEP>
                                <STLIST|INTERSECTION>
                            <TERM>
```

```
SEGMENT
  TRANSFORMATION          ::= SEGTRAN
                                <SOFTSEP>
                                <I:SEGID>
                                <SEP>
                                <TM:TRANMATRIX>
                              <TERM>

SEGMENT HIGHLIGHTING      ::= SEGHIGHL
                                <SOFTSEP>
                                <I:SEGID>
                                <SEP>
                                <NORMAL|HIGHL>
                              <TERM>

SEGMENT DISPLAY
  PRIORITY                ::= SEGDISPPRI
                                <SOFTSEP>
                                <I:SEGID>
                                <SEP>
                                <I:DISPLAYPRIORITY>
                              <TERM>

SEGMENT PICK PRIORITY     ::= SEGPICKPRI
                                <SOFTSEP>
                                <I:SEGID>
                                <SEP>
                                <I:PICKPRIORITY>
                              <TERM>
```

Page 30

Add the following at the end of clause 7:

Pick identifier:

```
NAME PRECISION:
        MININT      -32767
        MAXINT       32767
```

APPENDIX 2

PDAM TEXT OF CGM AMENDMENT 3

ISO 8632/Am.3

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer of Picture Description Information

Part 1

Functional Specification

0 Introduction

Page 3

Subclause 0.8, add the following new paragraph after the first paragraph:

ISO 8632-1:1987/Am.3:1990 (8632/Am.3) uses font concepts and the font architecture defined in ISO/IEC DIS 9541-1 for defining CGM references to fonts and font resources. The font properties of 9541-1 are adopted where appropriate to define CGM mechanisms to assist in font substitution between metafile interchanging parties. 8632/Am.3 includes from 9541-1 the minimum amount of description necessary to indicate the concepts and properties being incorporated from 9541 into CGM. Clause 3 contains a number of glossary definitions that are taken from and are identical to 9541-1.

2 References

Page 5

Clause 2, References add the following references:

CIE Publication 17-4 International Lighting Vocabulary, 1986 (4th edition)

CIE Publication 15-2 Colorimetry, 1986 (2nd Edition)

CIE Publication S002, Colorimetric Observers, 1986 (1st edition)

ISO/IEC/DIS 10036:1988, Information processing systems — Procedure for registration of glyph and glyph collection identifiers.

ISO/IEC/DIS 9541-1:1990, Information processing systems — Font information interchange, Part 1: Architecture.

3 Definitions and abbreviations

Page 6

Sub-clause 3.1, add or change the following definitions:

3.1.1 CIELAB: A colour model, defining an absolute colour space based on colour matching experiments, whose components are L (Lightness) and A, B (Chromaticity).

3.1.2 CIE uniform colour space: One CIE recommended uniform colour space, CIELAB, is allowed in the CGM. This colour space is a non-linear transformation of the CIE 1931 XYZ tristimulus space, into the perceptual attributes of brightness and Chroma. CIELAB closely approximates a uniform colour space over small distances, and provides an approximately uniform measure of perceived colour differences.

3.1.3 CMYK colour space: A colour space based on the subtractive colour mixture of Cyan (C), Magenta (M) and Yellow (Y) primaries with the inclusion of black (K).

3.1.4 colour component: One of the dimensions of a colour space.

3.1.5 colour model: A specification of a 3D colour coordinate system and a 3D subspace in the coordinate system within which each displayable colour is represented by a point. Some colour models include a fourth, redundant, dimension to allow the independent representation of black. For the purpose of ISO 8632 colour model refers to one of RGB, CIELAB, or CYMK.

3.1.6 colour selection mode: Indicator as to whether colour selection is to be direct (by specifying a colour value) or indexed (by specifying an index into a table of colour values). See COLOUR VALUE.

3.1.7 colour value: Value of the n-tuple of components describing a colour in a given colour model.

3.1.8 escapement: During the rendering of text strings onto a display, the movement of the current position on the presentation surface after a glyph representation is imaged.

3.1.9 escapement point: A glyph metric; a point in the glyph coordinate system, to which the current position on the presentation surface is usually translated, after the glyph representation is imaged.

3.1.10 font: A collection of glyph images having the same basic design, e.g., *Courier Bold Oblique*.

3.1.11 font family: A collection of fonts of common design, e.g., *Courier, Courier Bold, Courier Bold Oblique*.

3.1.12 font resource: A collection of glyph representations together with descriptive and font metric information which are relevant to the collection of glyph representations as a whole.

3.1.13 glyph: An identified abstract graphical symbol independent of any actual image.

3.1.14 pel: Photographic element; a term originally defined by CCITT to have size, color, and shape. As used in CGM it is distinguished from "cell" in that the CELL ARRAY element of CGM and the other graphics standards assigns a particular rendering requirement to cells. Like cells, pels have size relative to the coordinate space of the picture. Pels are distinguished from pixels in that the latter are indivisible device-dependent dots — their size is only defined relative to a specific device, not relative to the VDC picture.

3.1.15 posture: The extent to which the shape of a glyph or set of glyphs appear to incline, including any consequent design or form change.

3.1.16 RGB colour space: A colour space with colorimetric coordinates based on red, green and blue reference stimuli or primaries. The RGB values used are intensities. Colour values may be negative in certain areas outside the gamut defined by the RGB primaries.

3.1.17 reference colour model: Basic colour model within CGM relative to which relationships to specifiable colour models (RGB, CYMK, and CIELAB) are calibrated. The reference colour model is defined by the CIE 1931 standard colorimetric system (XYZ).

3.1.18 symbol: A graphical object which is included at some point in the metafile by reference, either to a definition internal to the metafile or to a symbol collection external to the metafile.

3.1.19 weight: The ratio of a glyph's or set of glyphs' stem width to font height.

4 Concepts

Page 10

Subclause 4.2, add the following at the end:

A compound path may be defined for drawing a compound line primitive and for displaying text strings along an arbitrary text path. A path is defined by line primitive elements occurring between BEGIN COMPOUND PATH and END COMPOUND PATH elements.

A compound clipping or shielding region may be defined by line and filled-area elements occurring between BEGIN PROTECTED REGION and END PROTECTED REGION elements.

A tiled pel array may be defined by pel array elements occurring between BEGIN TILED PEL ARRAY and END TILED PEL ARRAY.

A geometric pattern may be defined by graphical primitive and primitive attribute elements which occur between the elements BEGIN GEOMETRIC PATTERN and END GEOMETRIC PATTERN.

The exact list of elements which may occur in any of these definition states will be found in the State Table, Table 4.X.

Page 10

Subclause 4.3, add the following to the list of elements given in the first paragraph of this clause:

COLOUR MODEL
COLOUR CALIBRATION
FONT PROPERTIES
GLYPH MAPPING
SYMBOL LIBRARY LIST

Page 11

Sub Clause 4.3.2: Add the following new subclause:

4.3.2.6 Version_3 set

The Version_3 set may be used to indicate all elements in the Version_2 set and all the additional elements defined in ISO 8632-1:1987/Am.3:1990. The additional elements are:

BEGIN GEOMMETRIC PATTERN
END GEOMETRIC PATTERN
BEGIN COMPOUND PATH
END COMPOUND PATH
BEGIN PROTECTED REGION
END PROTECTED REGION
BEGIN TILED PEL ARRAY
END TILED PEL ARRAY
COLOUR MODEL
COLOUR CALIBRATION
FONT PROPERTIES
GLYPH MAPPING
SYMBOL LIBRARY LIST
PICTURE MAPPING
PROTECTED REGION INDICATOR

DELETE PROTECTED REGION
HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
POLYBEZIER
SYMBOL
BITONAL PEL ARRAY
PEL ARRAY
LINE & EDGE TYPE DEFINITION
HATCH STYLE DEFINITION
LINE CAP
LINE JOIN
LINE MITRE LIMIT
EDGE CAP
EDGE JOIN
EDGE MITRE LIMIT
TEXT SCORE TYPE
RESTRICTED TEXT TYPE
LINE TYPE CONTININUATION
LINE TYPE INITIAL OFFSET
EDGE TYPE CONTININUATION
EDGE TYPE INITIAL OFFSET
GEOMETRIC PATTERN EXTENT
INTERPOLATED INTERIOR DEFINITION
SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL HEIGHT
SYMBOL ORIENTATION
PEL ARRAY REFERENCE POINT
PEL ARRAY COMPRESSION METHOD
GENERALIZED PATH TEXT MODE

Page 11

Add the following after subclause 4.3.3:

4.3.4 Font List and Font Resources

ISO/DIS 9541-1 defines an architecture for font resources, but does not define or standardize applications' use of the information in a font resource — ranging from gross or aggregate properties such as font posture to very specific and detailed properties such as individual glyph metrics. A metafile generator (with its associated application) will be a user of such font resource information. The application, in defining a picture which contains text strings, has knowledge of the properties of the font resource. It makes use of these properties to format or layout strings of text so that the complete strings have the desired characteristics.

CGM is used to transmit such pictures from a generating application to an interpreting application, possibly remote in time and space and possibly of very different architecture and resource availability. The font facilities of CGM are designed to provide a font referencing mechanism. Font referencing is the process of identifying or characterizing a font resource. Referencing may include identification of a specific font by name, or provide sufficient descriptive information to permit identification of a suitable font or substitute.

The FONT LIST element of CGM allows the exact naming of a font resource. Such font resources may in the future be registered and given structured names under the mechanisms of ISO 9541. In the ideal case the metafile interpreter recognizes and has available the font resource named in the FONT LIST. For cases where the named font is not available to the interpreter, the CGM has elements (FONT PROPERTIES and GLYPH MAPPING) which allow generators to pass to interpreters additional descriptive information about desired

Concepts

fonts and font resources. An alternative font can be selected by an interpreter through this descriptive information if the specified one is not available.

Page 11

Add the following after subclause 4.3.4:

4.3.5 Font and Glyph Elements

The FONT PROPERTIES element can be used to guide selection of a best fit font if an exact match is not available on a specific device. The font properties which may appear are a subset of those in the Minimum Font Description Subset of ISO 9541. The element allows prioritization of the importance of the properties. In the case that a font named in the FONT LIST is not present, the prioritized properties instruct the interpreter of the relative importance of the various characteristics of the requested font. In some cases it may not even be desired to get a particular font, but rather any font with certain characteristics — boldness, presence of serif, etc. The FONT PROPERTIES element enables generators to specify such concepts. The use of this information by interpreters is not standardized.

ISO/IEC/DIS 10036 specifies a procedure and a registrar (registering authority) for registering typographic glyph collections. There currently is no standard that associates codes (i.e., character codes) with these glyphs. However the registrar — the Association for Font Information Interchange, or AFII — assigns a unique 4-byte integer indentifier with each glyph.

8632/Am.3 defines a means to access these registered glyph collections. The GLYPH MAPPING element associates the AFII 4-byte identifiers with single-byte or multi-byte codes. A set of such codes is defined as a collection, forming a locally defined character set for use within the metafile. The local character set is associated with an index, and within the body of the CGM the normal character set access and switching mechanisms (based upon and adopted from ISO 2022) may be used to access the AFII registered glyphs within CGM text strings.

NOTE — The glyph complement is a property of a font resource in the ISO 9541 font architecture When the separate mechanisms of 8632/Am 3 for font reference and glyph access are used there is potential for incompatibility between the specifications — the requested glyph complement may not be representable in the requested font This same situation pertains in ISO 8632 1987

page 12

Add after section 4.4.6:

4.4.7 Picture Mapping

The PICTURE MAPPING element specifies a 3x3 matrix which is applied to all the coordinates in a picture to produce an affine transformation as shown in figure XX. The mapping is applied after any segment transformations but before device viewport control. This is illustrated in Figure YY.

A 3x3 matrix M transforms a point (x,y) at level (A) in the pipeline to a point (x',y') at level (B) in the pipeline as follows:

$$\begin{vmatrix} a & b & c \end{vmatrix} = M \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

The point (x',y') is recovered from the vector (a,b,c) by:

$$x' = a/c$$
$$y' = b/c$$

NOTE — In the matrix M the third row should be [0 0 c]

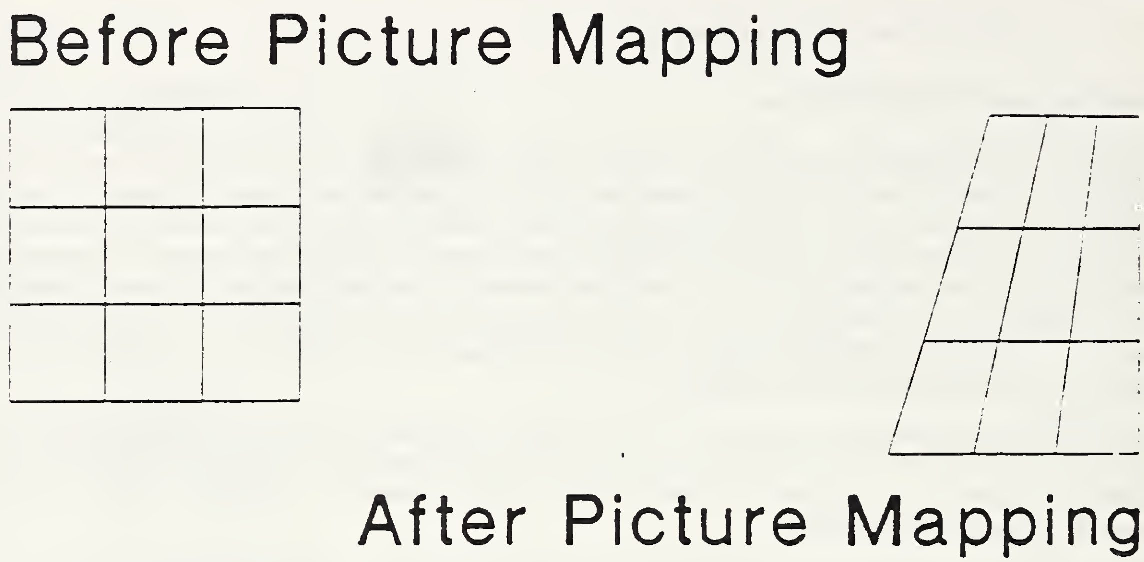


Figure XX. Example of the effect of PICTURE MAPPING.

```
 ---------------------
|   copy segment      |
|   transformation    |
 ---------------------
          |
          V
 ---------------------
|      segment        |
|   transformation    |
 ---------------------
          |................. (A)
          V
 ---------------------
|  picture mapping    |
|   transformation    |
 ---------------------
          |................. (B)
          V
 ---------------------
|     VDC to DC       |
|   transformation    |
 ---------------------
```

Figure YY. The order of transformations.

Concepts

Page 12

Subclause 4.4.2, first line, change:

direct (RGB) colour

to

direct colour

Page 14

Subclause 4.4.6, second paragraph, first line, change:

RGB

to

a direct colour

Page 15

After sub-clause 4.5.3, add:

4.5.X Compound Clipping and Shielding

The clipping and shielding elements consist of BEGIN PROTECTED REGION, END PROTECTED REGION, and PROTECTED REGION INDICATOR. The BEGIN/END elements are delimiter elements and the PROTECTED REGION INDICATOR element is a control element.

The concepts of clip and shield regions are complementary. The clipping process discards everything that is visually outside the clip region whereas the shielding process discards everything that is inside the shield region. Whether clipping and shielding are in effect is determined by the setting of the PROTECTED REGION INDICATOR.

Due to being able to define what amounts to closed figures for these regions, the clip region and shield regions may have "holes" within them. For example an annulus could be defined as the shield region, in which case it is the ring that is protected from drawing. The inner circle is not protected and may be drawn in. Locally the inner circle looks like a clip region, but in fact it is not, because drawing will happen in any area that is both outside of and outside of the surrounding shield region.

Page 15

Subclause 4.6, add the following to the list of graphical primitive elements:

HYPERBOLIC ARC
PARABOLIC ARC
BITONAL PEL ARRAY
PEL ARRAY
NON-UNIFORM B-SPLINE
POLYBEZIER
SYMBOL

Page 15

Subclause 4.6, add the following paragraph after the list of graphical primitive elements:

In addition, a tiled pel array compound graphical primitive may be defined by a sequence of PEL ARRAY and BITONAL PEL ARRAY elements between the BEGIN/END TILED PEL ARRAY delimiters.

Page 15

Subclause 4.6 add the following to the list of line elements:

HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
POLYBEZIER

Page 16

Subclause 4.6, before the last paragraph add:

The pel array primitive elements are:

PEL ARRAY
BITONAL PEL ARRAY

The single symbol primitive element is:

SYMBOL

Page 16

Subclause 4.6.1.1, change subclause to read the following:

4.6.1.1 Description. There are two general line elements — POLYLINE and DISJOINT POLYLINE — as well as line elements that define conic arcs — circular, elliptical, parabolic, and hyperbolic arcs — and elements that define spline curves.

Page 16

Subclause 4.6.1.1, change the end of the subclause:

HYPERBOLIC ARC:	generates a hyperbolic arc; the parameterization is described in 5.6.X, and the principles underlying the transformable parameterization are described in 4.X.Y.
PARABOLIC ARC:	generates a parabolic arc: the parameterization is described in 5.6.X+1, and the principles underlying the transformable parameterization are described in 4.X.Y.
NON-UNIFORM B-SPLINE:	generates a Non-Uniform B-Spline curve; the parameterization is described in 5.6.X, and the principles underlying the definition of the element are described in 4.X.Y.
POLYBEZIER:	generates a sequence of one or more Bezier curves: the parameterization is described in 5.6.X.

Page 16

Subclause 4.6.1.3, change the last sentence of the subclause to read:

The conic arc primitives (circular, elliptical, hyperbolic, and parabolic) and spline primitives (Bezier and Non-uniform B-splines)...

Page 18

Sub-clause 4.6.3.3, add the following text at the end of the sub-clause:

The GENERALIZED TEXT PATH MODE element selects the method for placing the text along the text path. When the mode is 'off' the text is displayed along the last text path defined by the BEGIN/END COMPOUND PATH elements.

When GENERALIZED TEXT PATH MODE is 'non-tangential' the characters are drawn along the text path but the character orientation vectors and axis are not rotated relative to the text path. When GENERALIZED TEXT PATH MODE is 'axis-tangential' the x-axis of the local character orientation axes is placed along the path and tangent to the path.

Examples of GENERALIZED TEXT PATH MODE are shown in Figure X.

Page 18

Subclause 4.6.4.1, change the second sentence of the subclause to read:

"In addition there are several elements that..."

Page 18

Subclause 4.6.4.3, 2nd paragraph, change the sentence to read:

The circular and elliptical fill primitives, as well as closed figure fill primitives incorporating such line primitives as the conic arc elements and spline curve elements...

Page 19

Add the following as subclause 4.6.5.1:

4.6.5.1 Pel Array Elements.

PEL ARRAY:	defines a rectangular raster image, either uncompressed or compressed according to one of a number of compression methods. The colours associated with the pels may either be bitonal or full colour, may be specified by either indexed or direct mode, and are specified according to the applicable colour precisions and modes.
BITONAL PEL ARRAY:	defines a rectangular raster image, either uncompressed or compressed according to one of a number of compression methods. Only two colours are used to define the image. Each pel is associated with one of the colour indexes 0 or 1, and the colour values associated with 0 and 1 are defined locally by each BITONAL PEL ARRAY element.
Tiled Pel Array	A tiled pel array is a compound raster image primitive, whose definition is delimited by the BEGIN/END TILED PEL ARRAY delimiter elements. Between the delimiter elements is a series of equally sized individual images or "tiles" which form a contiguous rectangular block. Each tile is

defined by a PEL ARRAY or BITONAL PEL ARRAY element. The first tile — the tile with pel array identifier 1 — is placed at the PEL ARRAY REFERENCE POINT, and subsequent tiles are placed at the tile position corresponding to their PID parameter. The tile positions are numbered as shown in figure X.

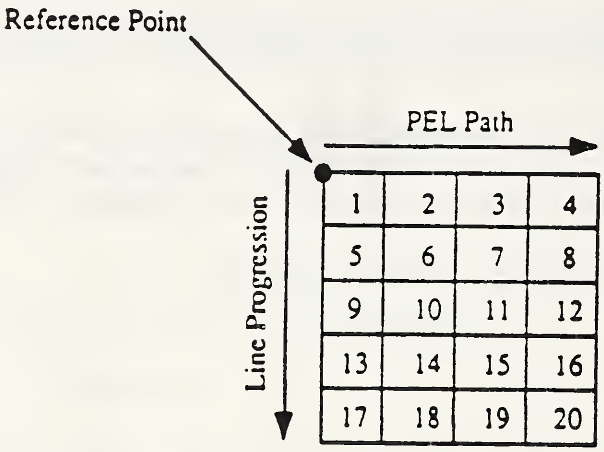


Figure X. Ordering and layout of tiles by index

4.6.5.1.1 Relationship to CELL ARRAY. A cell is a geometric entity just like lines and text. The concept of a pel falls between the concept of a cell and the concept of a device-dependent pixel. The assignment of device pixel colours to cells during metafile interpretation is specified by 8632-1:1987. The method of mapping of pels to device pixels is not mandated by this 8632 amendment. Cells are fully transformable by rotation, scaling, translation, and skewing. Pel array is always axis aligned and pels are always rectangular. Pels scale to the view surface but do not otherwise transform.

4.6.5.1.2 Allowable states for pel array elements. The pel array elements may appear in Picture Open State (POS) or Tiled Pel State (TPS). Tiled pel array may appear only in Picture Open State. The pel array elements may not appear in segments, geometric patterns definitions, or other similar states.

4.6.5.1.3 Compressed pel data. The pel colour data of the pel array elements is a compressed stream of pel colour specifiers. The datatype is Bitstream. For the BITONAL PEL ARRAY the Bitstream parameter consists of a sequence of 1-bit binary colour indexes which are compressed by the selected technique (the list of techniques includes 'bitmap', which is uncompressed). The resulting compressed binary data object is the parameter of the element. Each of the CGM encodings (Binary, Character, and Clear Text) defines a technique for representing and encoding the compressed binary data object.

4.6.5.1.4 Positioning. The position of a pel array element is defined by the PEL ARRAY REFERENCE POINT element. The reference point affects the position of all pel array elements that follow it in the metafile, until the next PEL ARRAY REFERENCE POINT element.

4.6.5.1.5 Tiling. The tiling mechanism specified is based on the Tiled Raster Interchange Format that has been developed for ISO 8613 Part 7. Definition of a tiled pel array is initiated by the BEGIN TILED PEL ARRAY delimiter element and terminated by the END TILED PEL ARRAY element. During tiled pel array definition subsequent pel array elements define individual tiles within the tiled image. The number of tiles is determined by the parameters of the BEGIN TILED PEL ARRAY element.

The number of tiles defined during tiled pel array definition must match the number indicated by the BEGIN TILED PEL ARRAY element. Annex D contains recommendations for the case that the tiles are missing. The tiling offset and size parameters defines the position of the actual image data within tile space, relative to the PEL ARRAY REFERENCE POINT (there may be parts of the "tile space" which contain no useful information and are simply artifacts of tiling).

Concepts

Page 19

Subclause 4.6.7, add the following after the subclause:

4.6.8 Hyperbolic Arc Element

The conjugate diameters parameterization of ellipses and elliptical arcs has the property of being transformable — the ellipse defined by the transformed parameter data is the transformed ellipse. The conjugate diameter parameterization has other useful properties as well.

For simplicity consider the the ellipse that is centered at the origin, and let P_1 and P_2 designate the endpoints of the conjugate diameters. Let M be the 2x2 matrix whose first column is P_1 and whose second column is P_2. The transformation M maps points on the unit circle centered at the origin ($x^2 + y^2 = 1$) onto the ellipse. The unit circle is referred to as the "canonical ellipse". If the ellipse is non-degenerate then M is non-singular, hence invertable, and M^{-1} maps points on the ellipse onto points on the unit circle centered at the origin. M maps the unit vectors $u_1=(1,0)$ and $u_2=(0,1)$ respectively onto P_1 and P_2. These principles generalize easily to ellipses which are not centered at the origin — there is a translation term in the mapping so that the transformation is not linear but is affine.

The CGM parameterization of the hyperbolic arc parallels that of the ellipse closely. The "canonical hyperbola" is defined by $x^2 - y^2 = 1$. It passes through the points u_1 and u_2 and has "center" (the point where the asymptotes cross) at the origin. Then for any non-degenerate hyperbola "centered" at the origin there is a linear transformation which maps the canonical hyperbola onto the given hyperbola. This transformation maps u_1 and u_2 respectively onto a pair of points P_1 and P_2. In this case P_1 is on the hyperbola but P_2 is not. At P_1 the tangent to the hyperbola is parallel to the line from the origin to P_2. The asymptotes of the hyperbola are parallel to the vectors $P_1 + P_2$ and $P_1 - P_2$. Points with such properties are referred to as the ***conjugate radius endpoint*** and the ***transverse radius endpoint***. It is such a pair of points (plus the centre point) which is used to parameterize the hyperbola in CGM.

As with the ellipse, if the matrix M is formed whose columns are the points P_1 and P_2 the this is the invertable transformation which maps points on the canonical hyperbola onto points on the given hyperbola (and whose inverse maps the given hyperbola onto the canonical hyperbola). Once again the generalization to hyperbolas whose center is not the origin is straight forward.

As with elliptical arcs, the start and end of the hyperbolic arc are parameterized by vectors from the center.

In both the case of the ellipse and the case of the hyperbola, the conjugate parameterizations can be derived from x-y implicit equations and vice-versa.

4.6.9 Parabolic Arc Element

The same principles are used to parameterize parabolic arcs, but the analogy is not quite as strong between parabolic arc and elliptical arc as it was between hyperbolic arc and elliptical arc. The parameterization is again in terms of a transformation of a "canonical parabola". In this case, the canonical parabola is $2(x + y) = (x-y)^2$ for $x \leq 1$ and $y \leq 1$. This parabolic arc is symmetric about the line $y=x$, starts at u_1, curves through the fourth quadrant, passes through the origin, curves through the second quadrant, and ends at u_2.

The general parabolic arc is parameterized by the endpoints of the arc, P_1 and P_2 and the intersection of the tangents to the arc at the endpoints. This intersection point is called the "center" of the parabolic arc, C. Define $V_1 = P_1 - C$ and $V_2 = P_2 - C$, and form the 2x3 matrix M whose first column consists of the components of V_1, second column consists of the components of V_2, and third column consists of the components of C. For non-degenerate parabolic arcs M is an affine transformation that maps points on the canonical parabolic arc onto points on the given parameterized parabolic arc.

4.6.9 Spline Curve Elements

The CGM provides two spline curve elements: non-uniform B-splines and Bezier curves.

4.6.9.1 Non-uniform B-splines

The CGM provides a non-uniform B-spline curve element with the capability to define both rational and non-rational B-splines of varying orders.

4.6.9.1.1 Parameterization. The non-uniform B-spline is parameterized by a spline order, a list of knots, an indictor selecting rational or non-rational, a list of control points, and parameter range limits defining the curve section to be drawn.

4.6.9.1.2 Mathematical Definition. The non-uniform B-spline is expressed parametrically in the form:

$$G(t) = \sum_{i=0}^{i=n-1} P_i B_i^k(t)$$

where:

n number of control points;
P_i control points (2D(x,y) or 3D(x,y,w))
B_i^k B-spline basis functions defined by degree k and knot vector T.

The degree, k, of the basis functions is one less than the order supplied with the primitives definition.

The number of spans in the B-spline function is $m = n-k$.

The knot vector consists of a non-decreasing sequence of real numbers $(T_{-k},\ldots,T_0,\ldots,T_m,\ldots,T_{m+k})$.

The curve itself is defined for the range $[T_0,T_m]$:

$$T_0 \leq t \leq T_m$$

and can be confined to the range $[T_{min},T_{max}]$:

T_{min} and T_{max} are specified as part of the non-uniform B-spline primitive.

Let $B_i^k(t,[T_{i-k},\ldots,T_{i+1}])$ represent the B-spline basis function of degree k supported by the interval $[T_{i-k},T_{i+1}]$. Following is a recursive expression for evaluating this basis function:

$$B_i^0(t,[T_i,T_{i+1}]) = \begin{cases} 1 & \text{if } T_i \leq t \leq T_{i+1} \\ 0 & otherwise \end{cases}$$

$$B_i^k(t,[T_{i-k},\ldots,T_{i+1}]) = \frac{(t - T_{i-k}) * B_{i-1}^{k-1}(t,[T_{i-k},\ldots,T_i])}{T_i - T_{i-k}} + \frac{(T_{i+1} - t) * B_i^{k-1}(t,[T_{i-k+1},\ldots,T_{i+1}])}{T_{i+1} - T_{i-k+1}}$$

In the case of multiple identical knot values some denominators evaluate to 0. In such cases, as part of the above definition the indeterminate quantity $\frac{0}{0}$ is considered to be 0.

4.6.9.2 Polybezier.

This element defines one or more cubic Bezier curves.

4.6.9.2.1 Parameterization. The polybezier is parameterized by a list of points. The point list is divided into consecutive sets of 4 points. Each set defines a single Bezier curve.

4.6.9.2.2 Geometric Concepts. If the points in a given 4-point set are designated $P_{0..P_3}$, then the defined Bezier curve goes from P_0 to P_3 P_1 and P_2 as control points. The defined curve starts at P_0 and at P_0 is tangent to the line segment from P_0 to P_1. The curve ends at P_3 and at P_3 is tangent to the line segment from P_2 to P_3. The curve lies entirely within the convex hull defined by the points.

The curve is defined by the cubic parametric equations

$$X(t) = A_x t^3 + B_x t^2 + C_x t + X_0$$
$$Y(t) = A_y t^3 + B_y t^2 + C_y t + Y_0$$

as t ranges from 0 to 1. The six coefficients A_x, B_x, C_x, A_y, B_y, C_y are defined by

Concepts

$$X_1 = X_0 + \frac{C_x}{3}$$

$$Y_1 = Y_0 + \frac{C_y}{3}$$

$$X_2 = X_1 + \frac{(C_x + B_x)}{3}$$

$$Y_2 = Y_1 + \frac{(C_y + B_y)}{3}$$

$$X_3 = X_0 + C_x + B_x + A_x$$

$$Y_3 = Y_0 + C_y + B_y + A_y$$

The individual Bezier curves, if there is more than one, are not implicitly connected. The N^{th} curve is connected to the $(N-1)^{th}$ curve only if the points P_{4N} and $P_{(4N-1)}$ are identical.

Page 20

Subclause 4.6, add the following new subclause:

4.6.X Symbol Elements

4.6.X.1 Description

8632/Am.3 defines mechanisms to access external symbol libraries and include their symbols in the metafile by reference. There is one symbol primtive element.

SYMBOL	generates a symbol which will be sized and oriented according to the symbol attributes and place with its reference point coinciding with the specified position point.

4.6.X.2 Attributes

The selection, sizing and placement of symbols is specified by the attribute elements SYMBOL HEIGHT, SYMBOL COLOUR, SYMBOL ORIENTATION, and SYMBOL LIBRARY INDEX.

Selection of the current symbol library from the list of available libraries is done by the SYMBOL LIBRARY INDEX element. The Metafile Descriptor element SYMBOL LIBRARY LIST associates index values with symbol library names. Access to symbol libraries and symbols is analagous to acces to text fonts and glyphs. The SYMBOL LIBRARY LIST associates the names of external libraries with indexes for internal reference, just as FONT LIST associates font names with internal indexes; SYMBOL LIBRARY INDEX selects the current symbol library, just as FONT INDEX selects the current font for text display; and SYMBOL selects the particular symbol and gives its position, just as the character code selects the glyph within positioned text strings.

The symbol coordinate system is illustrated in figure X. The symbol extent box is the design size of the symbol that will be used to set the size of the symbol. The symbol need not be entirely contained within the symbol extent box. Each symbol will have a reference point (though all symbols in a symbol library need not have the same reference point). The position point specified in the SYMBOL element is aligned with the symbol's reference point when placing a symbol.

The SYMBOL HEIGHT specifies the VDC size to which the design height of the symbol (the design distance between the top and the bottom of the symbol extent box) is to be scale for symbol display.

SYMBOL ORIENTATION specifies a symbol up vector and base vector, which define the orientation, skew, and distortion of the symbol.

4.6.X.9 Usage

The way in which software above the metafile generator and/or the metafile generator may use SYMBOL ORIENTATION is described. To generate the SYMBOL ORIENTATION and SYMBOL HEIGHT elements, a vector whose length is the symbol height and whose direction is the desired symbol up vector is created. A second vector is also created with the same length, whose direction is negative 90 degrees from the up vector. This pair of vectors may be transformed in the graphical pipeline before being passed to the metafile generator as the parameters for the SYMBOL ORIENTATION element. If the resultant vectors are not orthogonal, the symbol extent box becomes a parallelogram, and the symbol is skewed. If the vectors have different lengths, the aspect ratio inherent in the symbol design will be altered according to the ratio of the vector lengths. If the positive angle from the up vector to the base vector is less than 180°, the symbol is mirror imaged. The SYMBOL HEIGHT element may be derived from the length of the transformed up vector.

If an anisotropic transformation is in effect in the graphical pipeline preceeding the metafile generator, the symbol height must be recomputed by the metafile generator for each change in orientation. The SYMBOL HEIGHT and SYMBOL ORIENTATION are decoupled. Thus, to a metafile interpreter, the absolute lengths of the vectors in SYMBOL ORIENTATION are not significant; only their directions and the ratio of their lengths are significant.

It is not possible to directly specify the width of the displayed symbol. The displayed width is influenced by the SYMBOL HEIGHT and the SYMBOL ORIENTATION (which gives the distortion of aspect ratio of the displayed symbol relative to the design ratio), but a given width can only be requested precisely using these attributes in combination with knowledge of the design size and aspect ratio of the symbol. By contrast the symbol height can be requested directly.

To set the width of a symbol, the generator must set it as a percentage (ratio) of the height. If a given width is desired, and the height is of secondary importance, then the generator must use the design aspect ration to first calculate the height corresponding to the desired width. This gives the desired width at the design aspect ratio. If a modified aspect ratio is desired, then the lengths of the orientation vectors must be adjusted and the height adjusted so that at the specified height and distortion of aspect ratio the desired width results.

Page 21

Sub-clause 4.7, Table 1, add the following elements to the list of individual attribute elements:

LINE AND EDGE TYPE DEFINITION
HATCH STYLE DEFINITION
LINE CAP
LINE JOIN
LINE MITRE LIMIT
EDGE CAP
EDGE JOIN
EDGE MITRE LIMIT
FONT SCORE TYPE
RESTRICTED TEXT TYPE
GENERALIZED TEXT PATH MODE
LINE STYLE CONTINUATION
LINE STYLE INITIAL OFFSET
EDGE STYLE CONTINUATION
EDGE STYLE INITIAL OFFSET
INTERPOLATED INTERIOR DEFINITION
PEL ARRAY COMPRESSION METHOD
PEL ARRAY REFERENCE POINT
SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL HEIGHT
SYMBOL ORIENTATION

Concepts

Page 22

Sub-clause 4.7, Table 2, add the following elements to the Affected Primitives of LINE elements:

HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
POLYBEZIER

Page 23

4.7.4.1 a), add to the list of interior styles:

'interpolated'

Page 32

Sub-clause 4.7.6, add the following text after the third paragraph:

GENERALIZED PATH TEXT MODE has the possible values 'off', 'non-tangential', and 'axis-tangential'. If the mode is 'off' then the writing direction will be as specified by the TEXT PATH element — 'right', 'left', 'up', or 'down'.

NOTE — The four values of TEXT PATH define four special cases of the GENERALIZED TEXT PATH, with paths which are straight lines pointing from the text position point in four indicated directions. Mode 'off' is equivalent to mode 'non-tangential' combined with a straight-line compound text path pointing in one of the directions right, left, up, or down.

When TEXT PATH MODE is 'non-tangential' or 'axis-tangential' the string is displayed along the current text path as specified within the preceding BEGIN/END TEXT PATH elements. The orientation of the characters along the path will depend on the mode. If the mode is 'non-tangential' the characters are positioned along the path and oriented as per the character orientation vectors but the character orientation axes are not rotated — each character has the same orientation regardless of the path direction. If the mode is 'axis-tangential' the x-axis of the character orientation axes is tangent to the path at the character position — the orientation of each character depends upon the path direction at the character's placement point.

The characters are placed along the path starting at the point specified by the position parameter in the TEXT or RESTRICTED TEXT element. If this position is not the same as the initial point of the current compound line path, then the path is translated uniformly so that the initial point and the position parameter coincide. The resulting translated path is the path along which the text is drawn.

The characters are dimensioned according to the CHARACTER HEIGHT and CHARACTER EXPANSION FACTOR and are oriented according to CHARACTER ORIENTATION. The direction of character placement in the string relative to CHARACTER ORIENTATION is along the path defined within the scope of the preceding BEGIN TEXT PATH and END TEXT PATH elements. If the string length exceeds the length of the path, the characters of the string will continue to be placed along the path defined by a vector whose tail is the last point of the path and whose direction is the direction of the path at the last point.

Page 37

Sub-clause 4.7.6, add the following text at the end of the sub-clause:

The RESTRICTED TEXT TYPE element specifies the manner in which the string specified with the RESTRICTED TEXT primitive will be restricted to the restricted text box (parallelogram). Possible values are:

1: basic
2: boxed
3: isotropic
4: justified

These methods have the following effects:

basic: (as described in IS 8632 version 1) the text string is constrained not to exceed the text restriction box.

boxed: the baseline to capline distance of the text string exactly fills the text restriction box in the vertical direction and the width of the string exactly fits the box in the horizontal direction.

isotropic: the text string is displayed as large as possible within the text restriction box without altering the ratio of the height to the width of the string. The text string will exactly fill the text restriction box in either the horizontal or vertical direction and the characters will have the same proportions as if no adjustments had been made. The baseline-to-capline distance of the text is the measurement which is matched to the vertical dimension of the box.

justified: the text string exactly fits the text restriction box in the width (horizontal) direction (the direction specified by the character base vector of the CHARACTER ORIENTATION element) without changing the proportions of the characters. That is, the height of the characters and their aspect ratio (expansion factor) are not altered.

Page 38

Sub-clause 4.7.7, replace the sub-clause with the following text:

4.7.7 Colour attributes

The CGM provides the following colour models: RGB (the default), CIELAB and CMYK. The selection of one of these models is made in the Metafile Descriptor.

The RGB additive colour model is the default colour model. The RGB colour model uses a 3-tuple of values providing the normalized weights of the red (R), green (G) and blue (B) components of the desired colour.

One CIE recommended uniform colour space, CIELAB, is allowed in the CGM. This colour space is a non-linear transformation of the CIE 1931 XYZ tristimulus space, into the perceptual attributes of brightness and Chroma. CIELAB closely approximates a uniform colour space over small distances, and provides an approximately uniform measure of perceived colour differences.

The colour may be calibrated by defining a reference white value and associated calibration data. The calibration data specifies the position of RGB in the CIEXYZ colour space. XYZ values are related to RGB values by the following equations:

$$X = X_r R + X_g G + X_b B$$
$$Y = Y_r R + Y_g G + Y_b B$$
$$Z = Z_r R + Z_g G + Z_b B$$

The default values are those specified by SMPTE:

$$\begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = \begin{vmatrix} .394 & .365 & .192 \\ .212 & .701 & .087 \\ .019 & .112 & .958 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix}$$

NOTE — The white point specified by this transformation is D65 This is not consistent with the illuminant specified for reflection, however, it is aligned with current standards and practices

The CMYK colour model is based on the subtractive colour mixture of Cyan (C), Magenta (M) and Yellow (Y) primaries with the inclusion of black (K). This model is used primarily in the printing industry.

Concepts

The CGM provides two mechanisms for colour selection: 'direct' and 'indexed'. In 'direct' colour selection, the colour is defined by providing values for the normalized weights of the colour components for the selected colour model. In 'indexed' colour selection, the colour is defined by an index into a table of direct colour values. Selection of one of these mechanisms is done by the COLOUR SELECTION MODE element.

For 'indexed' colour selection, the COLOUR TABLE attribute element is provided for changing the contents of the colour table. This element may appear in the Picture Descriptor. It may also appear throughout the picture body, however the effect of changes in the colour table on any existing graphical primitive elements that use the affected indices is not addressed in ISO 8632.

NOTE — It is recommended that COLOUR TABLE be restricted to the Picture Descriptor

For direct colour specification, normalized weights for the colour components of the selected colour model are specified. For example, in the default situation, the red, green, and blue components of the desired colour. In the abstract, each component of the 3-tuple or 4-tuple is normalized to the continuous range of real numbers [0,1]; the normalization also has the property that any 3-tuple or 4-tuple with identical components represents equal weights of the colour components. For any given component, one end of the range indicates that none of that component is included, and the other end indicates that the maximum intensity of that component is included in the colour, with an infinite number of component values in between. For the RGB colour model, for example, (0,0,0) thus represents black, (1,1,1) represents white, and (x,x,x) with x between 0 and 1 represents greys.

There is a Metafile Descriptor element, COLOUR VALUE EXTENT, which allows metafile generators to specify the minimum and maximum metafile colour values; these will correspond with the abstract (0,0,0) and (1,1,1).

Page 98

Sub-clause 4.7.8, add the following text to the end of the description of 'hatch':

Hatch styles may be user defined in the CGM. The hatch styles may define attributes for the lines in the hatch. The colour of the lines in the hatch may be defined by the HATCH STYLE DEFINITION element and may be defined to be be the current fill colour.

Page 98

4.7.8, change the first sentence of the second paragraph from:

"The INTERIOR STYLE attribute selects one of five styles..."

to:

"The INTERIOR STYLE attribute selects one of the styles..."

Page 98

Sub-clause 4.7.8, add the following text to the end of the description of 'pattern':

A pattern may also be defined as a geometric pattern using graphical primitive elements and primtive attribute elements between the delimiter elements BEGIN GEOMETRIC PATTERN and END GEOMETRIC PATTERN.

Page 98

4.7.8, add to the list of interior style descriptions:

interpolated: fill the interior using the interpolated colour gradient defined by the INTERPOLATED INTERIOR DEFINITION element.

Page 98

Sub-clause 4.7.8, insert after the list of interior styles:

Hatch styles do not transform with the scaling and rotating transformations of CGM.

Page 98

After sub-clause 4.7.9, add:

4.7.X Compound Line

The BEGIN COMPOUND PATH and END COMPOUND PATH delimiter elements define a compound line when the mode parameter of the BEGIN COMPOUND PATH element is 'compound line'. These elements permit the definition of a line that consists of a number of distinct elements, such as straight lines and arcs, which is treated as if it were a single line element. Thus, for example, line style would apply without change or interruption past a straight line segment onto a following arc segment. Likewise, the ends of the various component elements of the compound line are not considered as line ends but rather as line joints. Line attributes may not change within a compound line.

4.7.X Compound Text Path

The BEGIN COMPOUND PATH and END COMPOUND PATH delimiter elements define a compound text path when the mode parameter of the BEGIN COMPOUND PATH element is 'text path'. Compound text path definition is identical to compound line definition, except the compound line is drawn and the compound text path serves as a reference path for laying out subsequent text strings (and is not drawn).

The compound text path permits arbitrary, complex placement of text. Each glyph in a text string is placed with its reference point and alignment according to a tangent to the compound text path. This implicit tangent is the logical base line for each character cell. If a glyph's reference point aligns with the junction of two line elements of the compound text path, the logical base line is the line perpendicular to the perpendicular bisector of the tangents of both elements, passing through the reference point. Positioning of subsequent glyphs is based upon the distance between glyphs assuming a straight base line, but wrapped along the generalized curve of the compound text path. If there is more text than path, the path for the excess text is the straight line described by the tangent at the end of the compound text path.

4.7.X Picture Composition

The picture composition elements are:

BEGIN PROTECTED REGION
END PROTECTED REGION
PROTECTED REGION INDICATOR
DELETE PROTECTED REGION

In addition, CLIP RECTANGLE and CLIP INDICATOR may be used for protecting certain rectangular areas of the drawing surface from graphical output.

Two methods of protection are available: clipping and shielding. The clipping process discards everything that would be drawn outside a specified region. The shielding process discards everything that is inside a specified region.

Concepts

Protected regions are identified by an index. Multiple regions may be active simultaneously. Protected regions are constructed by the same primitive elements as closed figures. The interior of a given protected region is defined in the same way as the interior of a closed figure. Regions which are constructed by line elements are closed by NEW REGION, END PROTECTED REGION, or any filled area element. If the endpoints and beginning points of subsequent line elements are not identical they are implicitly connected by a straight line.

Protected regions behave as do clip rectangles with respect to segment transformations — they transform by the segment and copy transformations.

If separate protected regions are simultaneously active, then the aggregate protected region is the union of the individual regions. The odd-even rule is used to determine the interior of a given region, but when two separate regions overlap the area of overlap is considered within the interior of the active aggregate region.

Page 41

Replace the Amendment 1 State Table, which follows the State Diagram, with:

	CGM Higher States (part 1 of 4)						
CGM Element	PCS	MDS	DR(1)	GSS	PDS	POS	LSS
BEGIN METAFILE (2)							
BEGIN PICTURE	X	X					
BEGIN PICTURE BODY					X		
END PICTURE						X	
BEGIN SEGMENT		X				X	
END SEGMENT				X		X	
BEGIN FIGURE				X	X	X	
END FIGURE							
END METAFILE	X	X					
+BEGIN GEOMETRIC PATTERN (3)			X	X	X	X	
+END GEOMETRIC PATTERN							
+BEGIN COMPOUND PATH				X		X	X
+END COMPOUND PATH							
+BEGIN PROTECTED REGION				X		X	X
+END PROTECTED REGION							
+BEGIN TILED PEL ARRAY						X	
+END TILED PEL ARRAY							
METAFILE VERSION		X					
METAFILE DESCRIPTION		X					
VDC TYPE		X					
INTEGER PRECISION		X					
REAL PRECISION		X					
INDEX PRECISION		X					
COLOUR PRECISION		X					
COLOUR INDEX PRECISION		X					
NAME PRECISION		X					
MAXIMUM COLOUR INDEX		X					
COLOUR VALUE EXTENT		X					
METAFILE ELEMENT LIST		X					
METAFILE DEFAULTS REPLACEMENT		X					
FONT LIST		X					
CHARACTER SET LIST		X					
CHARACTER CODING ANNOUNCER		X					
METAFILE CATEGORY		X					
MAXIMUM VDC EXTENT		X					
SEGMENT PRIORITY EXTENT		X					
+COLOUR MODEL		X					
+COLOUR CALIBRATION		X					
+FONT PROPERTIES		X					
+GLYPH MAPPING		X					
+SYMBOL LIBRARY LIST		X					

Concepts

	CGM Higher States (part 2 of 4)						
CGM Element	PCS	MDS	DR(1)	GSS	PDS	POS	LSS
SCALING MODE			X		X		
COLOUR SELECTION MODE			X	X	X	X	X
LINE WIDTH SPECIFICATION MODE			X	X	X	X	X
MARKER SIZE SPECIFICATION MODE			X	X	X	X	X
EDGE WIDTH SPECIFICATION MODE			X	X	X	X	X
VDC EXTENT			X		X		
BACKGROUND COLOUR			X		X		
DEVICE VIEWPORT			X		X		
DEVICE VIEWPORT MAPPING			X		X		
DEVICE VIEWPORT SPECIFICATION MODE			X		X		
LINE REPRESENTATION			X		X		
MARKER REPRESENTATION			X		X		
TEXT REPRESENTATION			X		X		
FILL REPRESENTATION			X		X		
EDGE REPRESENTATION			X		X		
+PICTURE MAPPING			X		X		
VDC INTEGER PRECISION			X	X		X	X
VDC REAL PRECISION			X	X		X	X
AUXILIARY COLOUR			X	X		X	X
TRANSPARENCY			X	X		X	X
CLIP RECTANGLE			X	X		X	X
CLIP INDICATOR			X	X		X	X
LINE CLIPPING MODE			X	X		X	X
MARKER CLIPPING MODE			X	X		X	X
EDGE CLIPPING MODE			X	X		X	X
NEW REGION							
SAVE PRIMITIVE CONTEXT				X		X	X
RESTORE PRIMITIVE CONTEXT				X		X	X
+PROTECTED REGION INDICATOR			X	X		X	X
+DELETE PROTECTED REGION				X		X	X

	CGM Higher States (part 3 of 4)						
CGM Element	PCS	MDS	DR(1)	GSS	PDS	POS	LSS
POLYLINE				X		X	X
DISJOINT POLYLINE				X		X	X
POLYMARKER				X		X	X
TEXT				X		X	X
RESTRICTED TEXT				X		X	X
APPEND TEXT							
POLYGON				X		X	X
POLYGON SET				X		X	X
CELL ARRAY				X		X	X
GDP				X		X	X
RECTANGLE				X		X	X
CIRCLE				X		X	X
CIRCULAR ARC 3 POINT				X		X	X
CIRCULAR ARC 3 POINT CLOSE				X		X	X
CIRCULAR ARC CENTRE				X		X	X
CIRCULAR ARC CENTRE CLOSE				X		X	X
ELLIPSE				X		X	X
ELLIPTICAL ARC				X		X	X
ELLIPTICAL ARC CLOSE				X		X	X
CIRCULAR ARC CENTRE REVERSED				X		X	X
CONNECTING EDGE							
+PARABOLIC ARC				X		X	X
+HYPERBOLIC ARC				X		X	X
+NON-UNIFORM B-SPLINE				X		X	X
+POLYBEZIER				X		X	X
+SYMBOL				X		X	X
+BITONAL PEL ARRAY						X	
+PEL ARRAY						X	
LINE BUNDLE INDEX			X	X		X	X
LINE TYPE			X	X		X	X
LINE WIDTH			X	X		X	X
LINE COLOUR			X	X		X	X
MARKER BUNDLE INDEX			X	X		X	X
MARKER TYPE			X	X		X	X
MARKER SIZE			X	X		X	X
MARKER COLOUR			X	X		X	X
TEXT BUNDLE INDEX			X	X		X	X
TEXT FONT INDEX			X	X		X	X
TEXT PRECISION			X	X		X	X
CHARACTER EXPANSION FACTOR			X	X		X	X
CHARACTER SPACING			X	X		X	X
TEXT COLOUR			X	X		X	X
CHARACTER HEIGHT			X	X		X	X
CHARACTER ORIENTATION			X	X		X	X
TEXT PATH			X	X		X	X
TEXT ALIGNMENT			X	X		X	X
CHARACTER SET INDEX			X	X		X	X
ALTERNATE CHARACTER SET INDEX			X	X		X	X

Concepts

	CGM Higher States (part 4 of 4)						
CGM Element	PCS	MDS	DR(1)	GSS	PDS	POS	LSS
FILL BUNDLE INDEX			X	X		X	X
INTERIOR STYLE			X	X		X	X
FILL COLOUR			X	X		X	X
HATCH INDEX			X	X		X	X
PATTERN INDEX			X	X		X	X
EDGE BUNDLE INDEX			X	X		X	X
EDGE TYPE			X	X		X	X
EDGE WIDTH			X	X		X	X
EDGE COLOUR			X	X		X	X
EDGE VISIBILITY			X	X		X	X
FILL REFERENCE POINT			X	X		X	X
PATTERN TABLE			X	X		X	X
COLOUR TABLE			X	X		X	X
ASPECT SOURCE FLAGS			X	X		X	X
PICK IDENTIFIER			X	X		X	X
+LINE & EDGE TYPE DEFINITION			X	X		X	X
+HATCH STYLE DEFINITION			X	X		X	X
+LINE CAP			X	X		X	X
+LINE JOIN			X	X		X	X
+LINE MITRE LIMIT			X	X		X	X
+EDGE CAP			X	X		X	X
+EDGE JOIN			X	X		X	X
+EDGE MITRE LIMIT			X	X		X	X
+FONT SCORE TYPE			X	X		X	X
+RESTRICTED TEXT TYPE			X	X		X	X
+LINE TYPE CONTININUATION			X	X		X	X
+LINE TYPE INITIAL OFFSET			X	X		X	X
+EDGE TYPE CONTININUATION			X	X		X	X
+EDGE TYPE INITIAL OFFSET			X	X		X	X
+GEOMETRIC PATTERN EXTENT							
+INTERPOLATED INTERIOR DEFINITION			X	X		X	X
+SYMBOL LIBRARY INDEX			X	X		X	X
+SYMBOL COLOUR			X	X		X	X
+SYMBOL HEIGHT			X	X		X	X
+SYMBOL ORIENTATION			X	X		X	X
+PEL ARRAY REFERENCE POINT			X	X		X	X
+PEL ARRAY COMPRESSION METHOD			X			X	
+GENERALIZED PATH TEXT MODE			X	X		X	X
ESCAPE	X	X	X	X	X	X	X
MESSAGE	X	X	X	X	X	X	X
APPLICATION DATA	X	X	X	X	X	X	X
COPY SEGMENT				X		X	X
INHERITANCE FILTER			X	X		X	X
CLIP INHERITANCE			X	X		X	X
SEGMENT TRANSFORMATION				X			X
SEGMENT HIGHLIGHTING				X			X
SEGMENT DISPLAY PRIORITY				X			X
SEGMENT PICK PRIORITY				X			X

	CGM Lower States (part 1 of 4)					
CGM Element	FOS	TOS	GPS	PHS	PRS	TPS
BEGIN METAFILE						
BEGIN PICTURE						
BEGIN PICTURE BODY						
END PICTURE						
BEGIN SEGMENT						
END SEGMENT						
BEGIN FIGURE						
END FIGURE	X					
END METAFILE						
+BEGIN GEOMETRIC PATTERN						
+END GEOMETRIC PATTERN			X			
+BEGIN COMPOUND PATH			X			
+END COMPOUND PATH				X		
+BEGIN PROTECTED REGION						
+END PROTECTED REGION					X	
+BEGIN TILED PEL ARRAY						
+END TILED PEL ARRAY						X
METAFILE VERSION						
METAFILE DESCRIPTION						
VDC TYPE						
INTEGER PRECISION						
REAL PRECISION						
INDEX PRECISION						
COLOUR PRECISION						
COLOUR INDEX PRECISION						
NAME PRECISION						
MAXIMUM COLOUR INDEX						
COLOUR VALUE EXTENT						
METAFILE ELEMENT LIST						
METAFILE DEFAULTS REPLACEMENT						
FONT LIST						
CHARACTER SET LIST						
CHARACTER CODING ANNOUNCER						
METAFILE CATEGORY						
MAXIMUM VDC EXTENT						
SEGMENT PRIORITY EXTENT						
+COLOUR MODEL						
+COLOUR CALIBRATION						
+FONT PROPERTIES						
+GLYPH MAPPING						
+SYMBOL LIBRARY LIST						

	CGM Lower States (part 2 of 4)					
CGM Element	FOS	TOS	GPS	PHS	PRS	TPS
SCALING MODE						
COLOUR SELECTION MODE						
LINE WIDTH SPECIFICATION MODE						
MARKER SIZE SPECIFICATION MODE						
EDGE WIDTH SPECIFICATION MODE						
VDC EXTENT						
BACKGROUND COLOUR						
DEVICE VIEWPORT						
DEVICE VIEWPORT MAPPING						
DEVICE VIEWPORT SPECIFICATION MODE						
LINE REPRESENTATION						
MARKER REPRESENTATION						
TEXT REPRESENTATION						
FILL REPRESENTATION						
EDGE REPRESENTATION						
+PICTURE MAPPING						
VDC INTEGER PRECISION	X					
VDC REAL PRECISION	X					
AUXILIARY COLOUR	X	X				
TRANSPARENCY	X	X				
CLIP RECTANGLE						
CLIP INDICATOR						
LINE CLIPPING MODE						
MARKER CLIPPING MODE						
EDGE CLIPPING MODE						
NEW REGION	X					
SAVE PRIMITIVE CONTEXT						
RESTORE PRIMITIVE CONTEXT						
+PROTECTED REGION INDICATOR						
+DELETE PROTECTED REGION						

	CGM Lower States (part 3 of 4)					
CGM Element	FOS	TOS	GPS	PHS	PRS	TPS
POLYLINE	X		X	X	X	
DISJOINT POLYLINE	X		X	X	X	
POLYMARKER			X			
TEXT			X			
RESTRICTED TEXT			X			
APPEND TEXT		X				
POLYGON	X		X		X	
POLYGON SET	X		X		X	
CELL ARRAY			X			
GDP	X		X	X	X	
RECTANGLE	X		X		X	
CIRCLE	X		X		X	
CIRCULAR ARC 3 POINT	X		X	X	X	
CIRC ARC 3 POINT CLOSE	X		X		X	
CIRCULAR ARC CENTRE	X		X	X	X	
CIRCULAR ARC CENTRE CLOSE	X		X		X	
ELLIPSE	X		X		X	
ELLIPTICAL ARC	X		X	X	X	
ELLIPTICAL ARC CLOSE	X		X		X	
CIRCULAR ARC CENTRE REVERSED	X		X	X	X	
CONNECTING EDGE	X					
+PARABOLIC ARC	X		X	X	X	
+HYPERBOLIC ARC	X		X	X	X	
+NON-UNIFORM B-SPLINE	X		X	X	X	
+POLYBEZIER	X		X	X	X	
+SYMBOL			X			
+BITONAL PEL ARRAY						X
+PEL ARRAY						X
LINE BUNDLE INDEX			X			
LINE TYPE			X			
LINE WIDTH			X			
LINE COLOUR			X			
MARKER BUNDLE INDEX			X			
MARKER TYPE			X			
MARKER SIZE			X			
MARKER COLOUR			X			
TEXT BUNDLE INDEX		X	X			
TEXT FONT INDEX		X	X			
TEXT PRECISION		X	X			
CHARACTER EXPANSION FACTOR		X	X			
CHARACTER SPACING		X	X			
TEXT COLOUR		X	X			
CHARACTER HEIGHT		X	X			
CHARACTER ORIENTATION			X			
TEXT PATH			X			
TEXT ALIGNMENT			X			
CHARACTER SET INDEX		X	X			
ALTTERNATE CHARACTER SET INDEX		X	X			

Concepts

	CGM Lower States (part 4 of 4)					
CGM Element	FOS	TOS	GPS	PHS	PRS	TPS
FILL BUNDLE INDEX			X			
INTERIOR STYLE			X			
FILL COLOUR			X			
HATCH INDEX			X			
PATTERN INDEX			X			
EDGE BUNDLE INDEX	X		X			
EDGE TYPE	X		X			
EDGE WIDTH	X		X			
EDGE COLOUR	X		X			
EDGE VISIBILITY	X		X			
FILL REFERENCE POINT			X			
PATTERN TABLE						
COLOUR TABLE						
ASPECT SOURCE FLAGS	X		X			
PICK IDENTIFIER						
+LINE & EDGE TYPE DEFINITION			X			
+HATCH STYLE DEFINITION			X			
+LINE CAP			X			
+LINE JOIN			X			
+LINE MITRE LIMIT			X			
+EDGE CAP	X		X			
+EDGE JOIN	X		X			
+EDGE MITRE LIMIT	X		X			
+FONT SCORE TYPE			X			
+RESTRICTED TEXT TYPE			X			
+GENERALIZED TEXT PATH MODE			X			
+LINE STYLE CONTININUATION			X			
+LINE STYLE INITIAL OFFSET			X			
+EDGE STYLE CONTININUATION	X		X			
+EDGE STYLE INITIAL OFFSET	X		X			
+GEOMETRIC PATTERN EXTENT			X			
+INTERPOLATED INTERIOR DEFINITION			X			
+SYMBOL LIBRARY INDEX			X			
+SYMBOL COLOUR			X			
+SYMBOL HEIGHT			X			
+SYMBOL ORIENTATION			X			
+PEL ARRAY REFERENCE POINT						X
+PEL ARRRAY COMPRESSION METHOD						X
+GENERALIZED TEXT PATH MODE			X			
ESCAPE	X	X	X	X	X	X
MESSAGE	X		X	X	X	
APPLICATION DATA	X		X	X	X	X
COPY SEGMENT			X			
INHERITANCE FILTER			X			
CLIP INHERITANCE			X			
SEGMENT TRANSFORMATION						
SEGMENT HIGHLIGHTING						
SEGMENT DISPLAY PRIORITY						
SEGMENT PICK PRIORITY						

Notes on the state tables:

1. Defaults replacement mode is not really a metafile state, but for implementation purposes it behaves as one and so has been included in this table.

2 The Metafile Closed State is not included in this table — BEGIN METAFILE is the only elements allowed in this state

3 The elements that are new with Amendment 3 have been marked with a "+" in these tables.

Higher States

PCS Picture Closed State
MDS Metafile Descripton State
DR Defaults Replacement Mode
GSS Global Segment State
PDS Picture Description State
POS Picture Open State
LSS Local Segment State

Lower States

FOS Figure Open State
TOS Text Open State
GPS Geometric Pattern State
PHS Path State
PRS Protected Region State
TPS Tiled Pel State

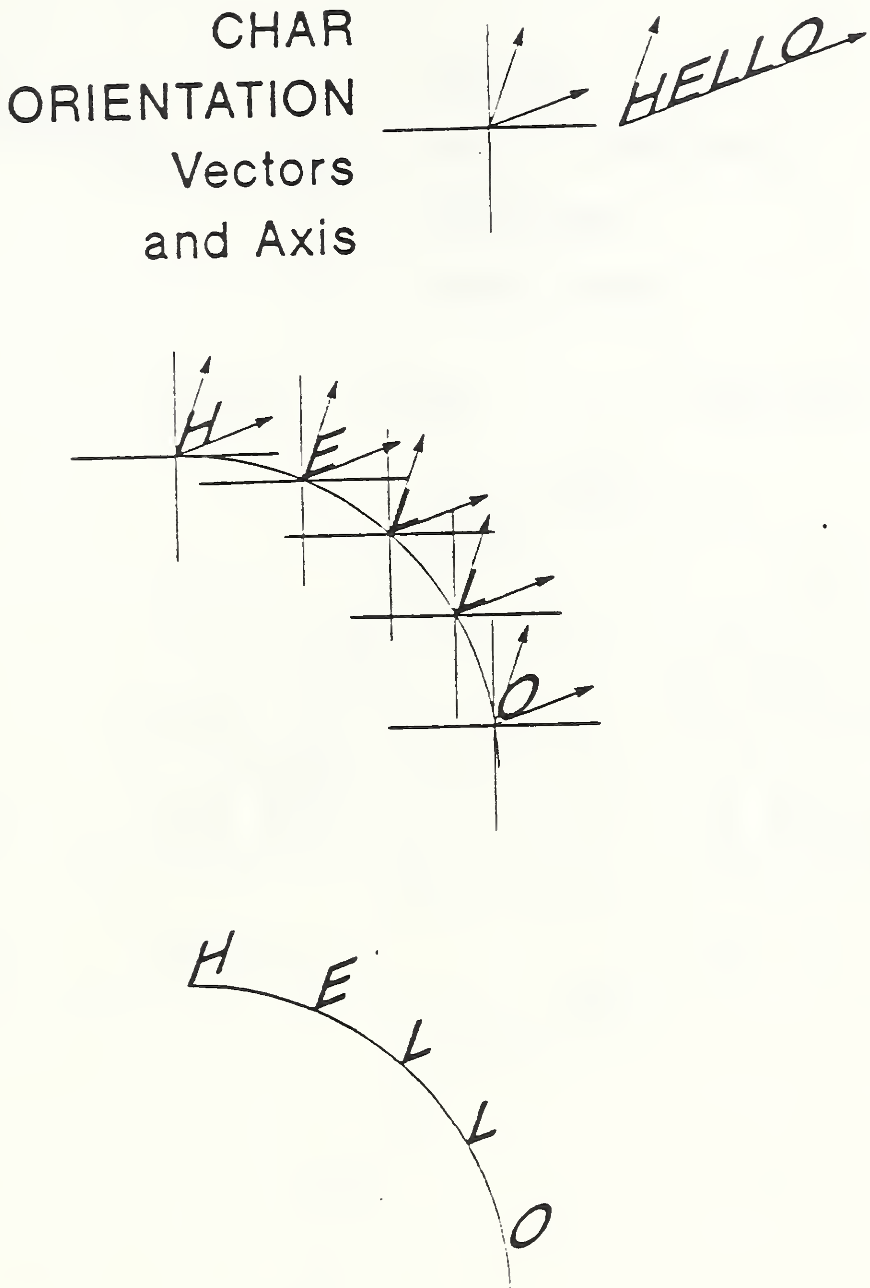


Figure X. (a) Examples of GENERALIZED TEXT PATH MODE, non-tangential

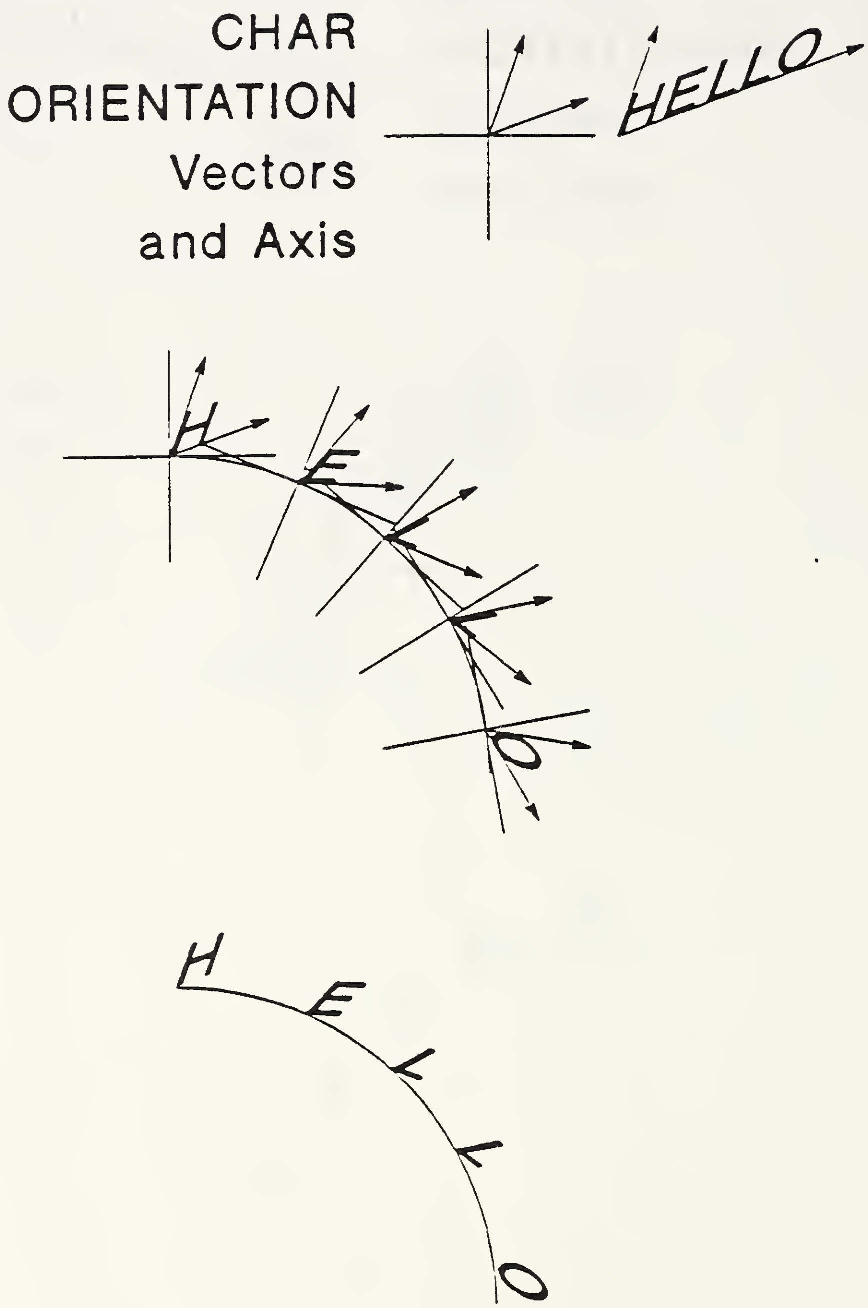


Figure X. (b) Examples of GENERALIZED TEXT PATH MODE, axis-tangential

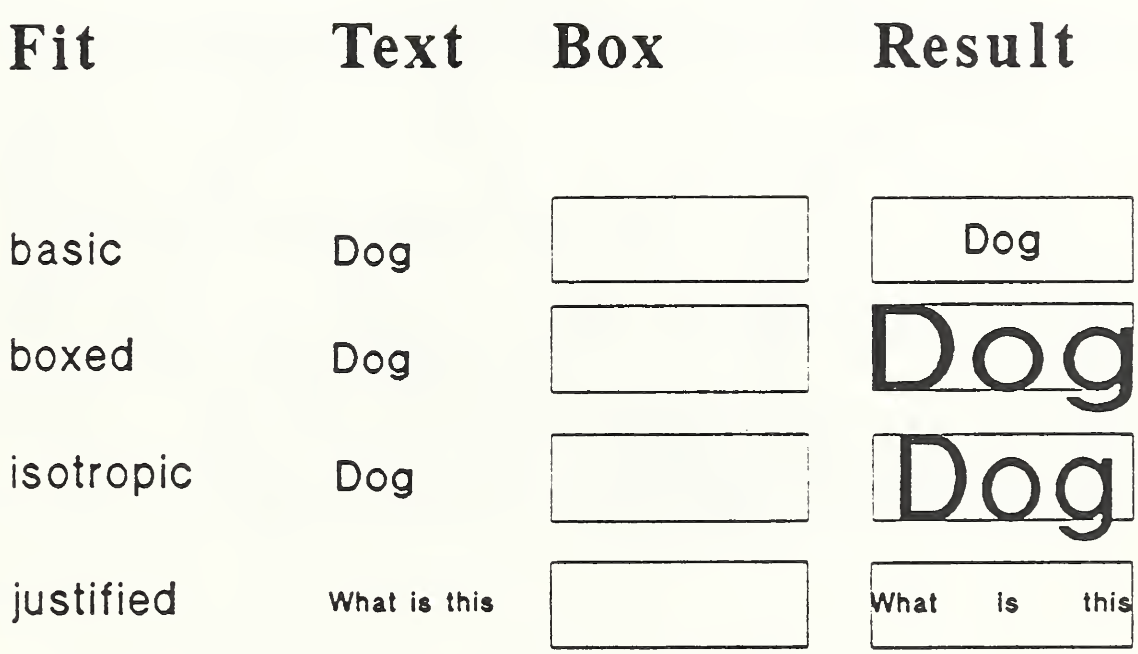


Figure X. Examples of RESTRICTED TEXT TYPE.

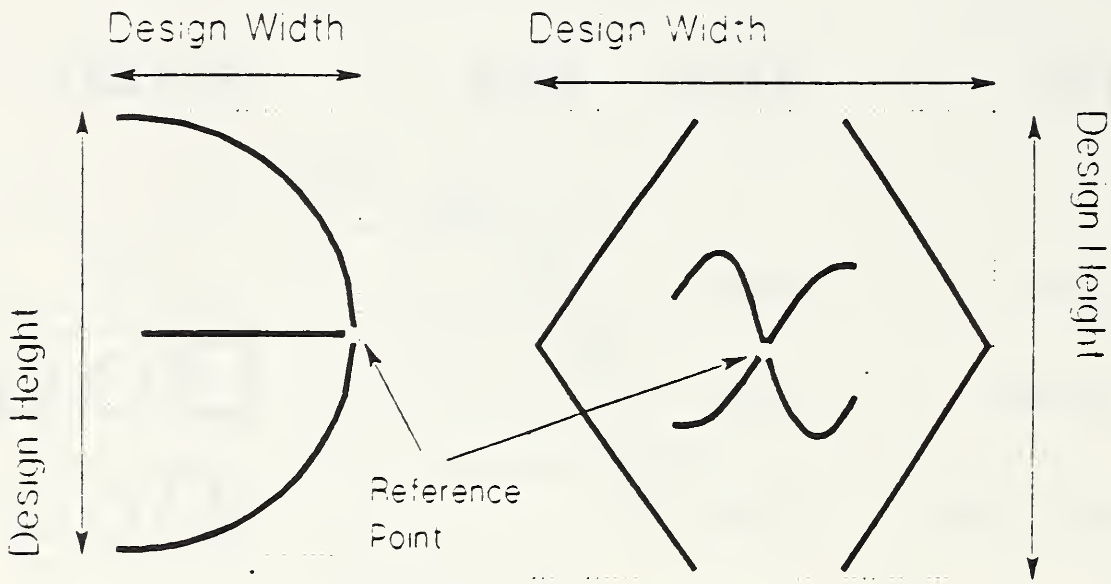


Figure X. Symbol coordinate system.

5 Elements

Page 42

Subclause 5.1, table of data type abbreviations, replace

CD	Colour Direct	three-tuple of non-negative real values for red, green, blue colour intensities.

with

CD	Colour Direct	three-tuple or four-tuple of non-negative values for colour definition within one of the supported colour models.

Page 42

Subclause 5.1, table of data type abbreviations, add:

OD	Octet	an unsigned integer in the range 0..255, always represented at fixed precision equivalent to 8 binary bits.
BS	Bitstream	a binary data object, given an encoding-dependent representation in each of the three encodings (8632/2, 8632/3, 8632/4), which consists of a compressed stream of the binary representations of other CGM datatypes (e.g., colours), compressed according to one of a number of standardized techniques defined in this part of CGM.

Page 42

Subclause 5.2, add the following Delimiter Elements to the subsection:

5.2.X BEGIN COMPOUND PATH

Parameters:

path type (one of: text path, compound line) (E)

Description:

If the path type is 'compound line' the definition of a compound line entity begins. The compound line entity will have consistent line attributes and will be treated as a single line primitive. Line attributes may not be changed while constructing a compound line.

If the mode is 'text path', the definition of an entity begins that will provide the path along which a text string will be drawn. The display of text along the defined text path is as described in clause 4.

References:

5.2.X END COMPOUND PATH

Parameters:

None

Description:

END COMPOUND PATH delimits the end of a compound line or text path definition.

References:

5.2.X BEGIN PROTECTED REGION

Parameters:

region index (ix)
region type (one of: clip, shield)

Description:

Line and fill primitives which are present between the BEGIN PROTECTED REGION and END PROTECTED REGION are used to construct a protected region. The region is used either for clipping or for shielding, as specified in the region type parameter. The defined region is associated with the region index parameter, by which it may subsequently be referenced by clipping and shielding control elements. Once defined a region index may not be redefined until the associated region has been deleted.

References:

5.2.X END PROTECTED REGION

Parameters:

None

Description:

END PROTECTED REGION delimits the end of a protected region definition.

References:

5.2.X BEGIN TILED PEL ARRAY

Elements

Parameters:

tiled pel array dimensions (2I)
tiling offset (2I)
image size (2I)

Description:

A tiled pel array image is defined as follows:

The tiled pel array dimensions parameter consists of two positive integers corresponding to the number of tiles in the direction of the Pel Path and Line Progression parameters respectively, of the pel array elements comprising the tiled array. The product of the two integers is the total number of tiles contained in the tiled pel array.

The number of pels per tile line equals the number of pels per line of the pel array elements comprising the tiled array. The number tile lines equals the number of tile lines of the pel array elements comprising the tiled array.

All PEL ARRAY and BITONAL PEL ARRAY elements contained in a tiled pel array are constrained to have identical dimensions and identical directions of progression of pels and lines.

When laid out, all of the tiles in the tiled pel array define a rectangular subregion of VDC space — a "tiling space". The actual graphical image may not (in fact in large tiled images likely will not) occupy the full rectangle. The image offset and image size parameters specify the rectangle within the tiling space which is actually occupied by the image. Image offset is measured from the PEL ARRAY REFERENCE POINT. Both parameters are measured in pels. Both parameters contain two components. The first is measured in the pel path direction and the second is measured in the line progression direction. These parameters are purely informative, to help interpreters locate and efficiently process the significant part of the tiled pel array.

References:

5.2.X END TILED PEL ARRAY

Parameters:

None

Description:

This element terminates the definition of tiled pel array which was commenced by BEGIN TILED PEL ARRAY.

References:

5.2.X BEGIN GEOMETRIC PATTERN

Parameters:

pattern table index (IX)

Description:

The definition of the representation of the specified pattern table index is begun. The pattern definition is comprised of a sequence of graphical primitives and attributes. The elements allowable during the pattern definition are specified in table XX, clause 4.X. The pattern definition is terminated by the element END GEOMETRIC PATTERN.

Legal values of the pattern table index are positive integers. Negative values of pattern indices are reserved for registration.

When INTERIOR STYLE is 'pattern' and the current PATTERN INDEX corresponds to a geometric pattern, then that geometric pattern is mapped onto the interior of a filled area as described in the PATTERN SIZE element.

NOTE — Conceptually there is a single pattern table containing both raster patterns — defined by PATTERN TABLE — and geometric patterns — defined by BEGIN/END GEOMETRIC PATTERN. This in no way constrains implementations to actually using a single table to handle the different pattern types.

NOTE — Pattern indices are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a geometric pattern has been approved by the ISO Subcommittee on Computer Graphics, the pattern index value will be assigned by the Registration Authority.

References:

5.2.X END GEOMETRIC PATTERN

Parameters:

None

Description:

The definition of the geometric pattern which was commenced by the preceding BEGIN GEOMETRIC PATTERN is terminated.

References:

Page 47

Subclause 5.3.1, METAFILE VERSION, add to the end of the subclause:

The CGM as defined in ISO 8632/1-1987/Am.3 is version three (3).

Pages 47-48

Subclause 5.3.7, COLOUR PRECISION, replace the second paragraph as follows:

Although the form of the parameter is encoding dependent, the parameter is a single specification that applies to each or all of the three or four components of parameters of type CD. The precisions of the individual components are not independently and differently specifiable by this element.

Elements

Pages 48-49

Subclause 5.3.10, COLOUR VALUE EXTENT replace the description as follows:

The parameters represent an extent which bounds the direct colour values that will be encountered in the metafile. It need not represent the exact extent of colour model contained in the metafile.

Page 50

Subclause 5.3.10, COLOUR VALUE EXTENT, replace the second paragraph of the description by:

The minimum and maximum values are a 3-tuple or 4-tuple giving the colour components corresponding to the normalized colour space, zero to one for each component. The values given will depend upon the colour model selected for use in the metafile.

Page 55

Subclause 5.3, add the following Metafile Descriptor Elements:

5.3.X COLOUR MODEL

Parameters:

colour model indicator (IX)

Description:

The colour model of the metafile is selected. Standardized values include:

1: RGB
2: CIELAB
3: CMYK

Colour representation by any of the following colour models is supported: RGB, CIELAB, CYMK.

Only one colour model may be used within a metafile. The method may be defaulted or explicitly set with the COLOUR MODEL element. All occurrences of colour-setting elements (AUXILIARY COLOUR, LINE COLOUR, MARKER COLOUR, FILL COLOUR, EDGE COLOUR, TEXT COLOUR), representation setting elements (COLOUR TABLE, PATTERN TABLE, LINE REPRESENTATION, MARKER REPRESENTATION, TEXT REPRESENTATION, FILL AREA REPRESENTATION), the colour lists of CELL ARRAY and PEL ARRAY, and any other place where a direct colour value may appear shall be in the selected colour model. If used, COLOUR MODEL shall be in the Metafile Descriptor, after BEGIN METAFILE and before the first BEGIN PICTURE.

References:
4.7.7

5.3.X FONT PROPERTIES

Parameters:

list of triples of (property name, property value, priority) (n[IX,{S|IX|R},I])

where property names must be one of:

font index (IX)
typeface name (S)
font family (S)
typeface design group (3IX)
posture (IX)
posture angle (R)
weight (IX)
proportional width (IX)
structure (S)

The property names above are type index; the datatypes in parentheses are the types for the property value associated with the name.

Description:

The priority parameter indicates the relative importance of the property for font substitution. The sum of all priorities is normalized to 1.0 and the relative priorities are computed as a fraction of 1.0. If, for example, no substitution is permissible, then the font index (which corresponds to a font name, defined by the FONT LIST element) could be given priority 10 and all other property priority 0. If, on the other hand, all that matters is a bold serif font, then weight and design group could be given priority 10 and all others priority 0.

The property names which may be referenced are from ISO/IEC/DIS 9541. Note that the font name itself (referenced by font index), which subsumes all other properties, is one of the properties.

The priotities given to the font properties provide guidance to the interpreter so as to enable rational font matching in the event of the inability to exactly match a font from the font name specified in the FONT LIST element. The priorities do not imply any particular font matching strategy, but do provide the means for generators to indicate relative importance of the various font properties.

Font index. The font number corresponds to the font index which is implicitly defined by the FONT LIST element.

Typeface name. The typeface name is generally the industrial name of the typeface, as normally used in the type industry. The typeface name is redundant with the font name (in the FONT LIST) and provides no additional information.

Font family. The name of the font family, for example *Courier.*

Typeface design group. The typeface design group consists of three components: the typeface general class, the typeface subclass, and the typeface specific group, as defined in ISO/IEC/DIS 9541-1:1990. The typeface general class is the most general grouping of fonts with similar characteristics. Typeface sub-classes are groupings that identify the less general characteristics and start to categorize typefaces into similar designs. Typeface specific groups are typeface groupings with very distinct and unique characteristics. Typefaces categorized to the typeface specific group level start to show similar characteristics that makes them reasonably eligible to be substituted for each other. The assigned fonts groups, and their properties, are defined by the normative annex A of ISO/DIS 9541-1. The three components are each assigned a value in the range 0..255 In annex A of 9541-1 a typeface design group specification looks like x.y.z, with each of x, y, and z in the range 0..255.

NOTE — The value for the posture angle property will normally be defined in the first and second quadrants of the glyph coordinate system, in the range 75° to 105°. For Latin fonts, the italic and oblique postures usually will be less than 90°

Posture. The posture of a font may be one of the following:

Elements

0: not applicable;
1: upright;
2: oblique — upright design slanted in the direction of the nominal escapement with no design or form change;
3: back slanted oblique — upright design slanted in the direction opposite of the nominal escapement with no design or form change;
4: italic — slanted in the direction of the nominal escapement with a change in design or form;
5: back slanted italic — italic design slanted in the direction opposite of the nominal escapement;
6: other.

Posture angle. The posture angle of a font is a real number representing an angle within the range of -360..+360 exclusive, the nominal angle of the posture of the typeface design group.

Weight. The font weight is a measure ofthe boldness of the font. Assigned values are:

0: not applicable;
1: ultra light (lowest ratio of glyph stem width to font height);
2: extra light
3: light
4: semi light
5: medium
6: semi bold
7: bold
8: extra bold
9: ultra bold (highest ratio of glyph stem width to font height);

Proportionate width. The proportionate width is an indication of the relative ratio of character height to character width, and may be one of the following:

0: not applicable;
1: ultra condensed (lowest ratio of glyph width to font height);
2: extra condensed;
3: condensed;
4: semi condensed;
5: medium;
6: semi expanded;
7: expanded;
8: extra expanded;
9: ultra expanded (highest ratio of glyph width to font height);

References:

5.3.X GLYPH MAPPING

Parameters:

character set index (IX)
basis set (S)
octets per code (I)
list of pairs of (code, glyph name) (n[BmOC,4OC])

where the number of octets that represent each code (mOC) is equal to the octets per code parameter.

Description:

A character set is defined for use in the metafile. The character set index can be used in CHARACTER SET INDEX and ALTERNATE CHARACTER SET INDEX elements. An index used in this element cannot also be declared in a CHARACTER SET LIST element. Each code in the defined character set will contain the number of octets indicated in the octets per code parameter. The basis set provides a default set of glyphs to use with any codes that are not assigned values by this element. The string that specifies the basis set is a designation sequence tail as defined for the CHARACTER SET LIST element (5.3.14). The glyph-to-code mapping is established by the list of pairs of codes and glyph names. Each item in the list associates a code with a glyph. Each glyph name is a 4-octet glyph identifier registered by the ISO Glyph Registration Authority, AFII.

References:

5.3.X COLOUR CALIBRATION

Parameters:

reference white value (Xn Yn Zn) (3R)
calibration data (3x3 matrix)

> Xr Xg Xb
> Yr Yg Yb
> Zr Zg Zb
> (9R)

Description:

Colour calibration allows the specification of the reference white value in CIEXYZ reference colour space. The calibration data specifies the position of RGB in the CIEXYZ colour space. The calibration data supplies the information which defines the transformation from the colour space values to the reference space. The Reference White Value for CIELAB specifies the XYZ values (Xn Yn Zn) of the reference white used in the equations that convert from CIELAB to XYZ colour space. For the RGB colour space the 3x3 matrix of calibration data specifies the values used to position the Red, Green, and Blue colours in the XYZ colour space.

References:

5.3.X SYMBOL LIBRARY LIST

Parameters:

symbol library names (nS)

Description:

This element permits selection of named symbol libraries via SYMBOL LIBRARY INDEX. The first symbol library defined in the symbol library list is assigned to index 1, the second to index 2, and so on.

Elements

NOTE — The strings may contain registered names or private names. Use of the former is recommended for metafile transportability, because registration ensures unique naming of symbol libraries

NOTE — Symbol Libraries are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a symbol library has been approved by the ISO working Group on Computer Graphics, the symbol library name will be assigned by the Registration Authority.

Page 56

Subclause 5.4.2, first paragraph of description:

Change "red, green, and blue" to "direct"

Page 56

Subclause 5.4, Picture Descriptor Elements, add new element:

5.4.X PICTURE MAPPING

Parameters:

mapping matrix (3x3xR)

Description:

The PICTURE MAPPING element specifies a 3x3 matrix which is applied to all the coordinates in a picture. The mapping is applied after any segment transformations but before device viewport control.

The transformation is applied to a point (x,y) as follows:

$$\begin{vmatrix} a & b & c \end{vmatrix} = M \begin{vmatrix} x \\ y \\ 1 \end{vmatrix}$$

The point (x',y') is recovered from the vector (a b c) by:

$$x' = a/c$$
$$y' = b/c$$

NOTE — The defined transformation will not necessarily be linear or affine. It will be in the case that the third row of M is (0 0 1)

NOTE — The first two elements of the last column of M are actually VDC, and the first two numbers of the last row are something like reciprocal VDC. For the purpose of this element, all values are embedded in real space and considered as reals.

References:

Page 58

Subclause 5.4.7, BACKGROUND COLOUR, first line of second paragraph of description:

Change "RGB" into "a direct colour value"

Page 61

Subclause 5.5, add the following new Control Elements to the end of the subclause:

5.5.X PROTECTED REGION INDICATOR

Parameters:

region index (IX)
region indicator (one of: off, on) (E)

Description:

The protected region indicator determines whether the clipping or shielding region associated with the given index is active and used respectively for clipping or shielding. It is independent of CLIP INDICATOR, which affects only the use of CLIP RECTANGLE.

5.5.X DELETE PROTECTED REGION

Parameters:

region index (IX)

Description:

A previously defined protected region is deleted from the list of available defined clip and shield regions. Once deleted it may no longer be referenced by a PROTECTED REGION INDICATOR. The index may be reused to define a new protected region.

References:

Page 63

Subclause 5.6.4 TEXT, add the following at the end of the second paragraph of the description:

If GENERALIZED TEXT PATH MODE is 'off', then text is positioned relative to the position point of the TEXT element as described in clause 4. If GENERALIZED TEXT PATH MODE is 'non-tangential' or 'axis-tangential' the BEGIN/END TEXT PATH element specifies the path the text string is to follow, and the method of orienting characters along the path is defined by the mode.

Page 64

Subclause 5.6.5: RESTRICTED TEXT, add the following at the end of the first paragraph of the description:

The RESTRICTED TEXT TYPE specifies how the string is positioned within the parallelogram.

Elements

Page 64

Subclause 5.6.5 RESTRICTED TEXT, add the following at the end of the third paragraph of the description:

If GENERALIZED TEXT PATH MODE is 'off', then text is positioned relative to the position point of the TEXT element as described in clause 4. If GENERALIZED TEXT PATH MODE is 'non-tangential' or 'axis-tangential' the BEGIN/END TEXT PATH element specifies the path the text string is to follow, and the method of orienting characters along the path is defined by the mode.

Page 64

Subclause 5.6.5 RESTRICTED TEXT, add the following text after the fifth paragraph of the description:

These attributes may be altered for RESTRICTED TEXT. The variation will depend upon the selected RESTRICTED TEXT TYPE.

Subclause 5.6.6, APPEND TEXT add the following at the end of the second paragraph of the description:

If GENERALIZED TEXT PATH MODE is 'off', then text is positioned relative to the position point of the TEXT element as described in clause 4. If GENERALIZED TEXT PATH MODE is 'non-tangential' or 'axis-tangential' the BEGIN/END TEXT PATH element specifies the path the text string is to follow, and the method of orienting characters along the path is defined by the mode.

Page 65

Subclause 5.6.6: APPEND TEXT, add the following at the end of the second paragraph of the description:

If GENERALIZED TEXT PATH MODE is 'off', then text is positioned relative to the position point of the TEXT element as described in clause 4. If GENERALIZED TEXT PATH MODE is 'non-tangential' or 'axis-tangential' the BEGIN/END TEXT PATH element specifies the path the text string is to follow, and the method of orienting characters along the path is defined by the mode.

Page 68

Subclause 5.6.9, add the following at the end of the third paragraph of the description:

Note that COLOUR PRECISION only applies to direct colour model values.

Page 77

Subclause 5.6, add the following Graphical Primitive Elements:

5.6.X HYPERBOLIC ARC

Parameters:

centre point (P)
transverse radius endpoint (P)
conjugate radius endpoint (P)
start vector (2VDC)
end vector (2VDC)

Description:

A hyperbolic arc is defined. The asymptotes of the full hyperbola pass through the centre point and are parallel to two vectors defined by the sum and difference of the vectors from the centre to the transverse and conjugate radius endpoints. The complete hyperbola passes through the transverse radius endpoint and is tangent there to the vector from the centre point to the conjugate radius endpoint. The defined arc is a finite arc starting and ending at the points where the rays from the centre in the directions of the start and end vectors intersect the complete hyperbola. See clause 4 for further discussion of the geometric significance of the parameterization and details of rendering of hyperbolic arcs.

See annex D for translation between this parametization and the x-y implicit equations for hyperbolic arcs.

References:

5.6.X PARABOLIC ARC

Parameters:

tangent intersection point (P)
start point (P)
end point (P)

Description:

A parabolic arc is defined. A parabolic arc is drawn from start to end point. The tangents to the arc at the endpoints intersect at the tangent intersection point. See clause 4 for further discussion of the geometric significance of the parameterization and details of rendering of hyperbolic arcs.

See annex D for translation between this parametization and the x-y implicit equations for parabolic arcs.

5.6.X NON-UNIFORM B-SPLINE

Parameters:

spline order (I)
list of knots (nR)
rationality (one of: rational, non-rational) (E)
control points (mP)
parameter start value (R)
parameter end value (R)

Elements

Description:

The spline order must be a positive integer. The knot sequence must form a non-decreasing sequence of numbers.

The rationality parameter may have the enumerated value 'rational' or 'non-rational'. When 'rational' is specified, the control points must be specified as homogeneous VDC coordinates, with the restriction that the third coordinate be greater than zero. For 'non-rational' types, the control points are ordinary VDC coordinates.

The number of control points must be at least as large as the spline order. The sum of the number of control points and the spline order must equal the number of knots.

The parameter start and end values specify over what range of the parameter the B-spline curve is evaluated. The start value must be less than the end value. The start value must be greater than or equal to the *order*-th know value. The end value must be less than or equal to the *(k+1-order)*-th knot value (where k is the number of knots).

When an element of this type is interpreted, a non-uniform B-spline curve is generated for parameter values between the parameter start value and parameter end value.

References:
4.x

5.6.X POLYBEZIER

Parameters:

point list (4nP)

Description:

This element defines one or more cubic Bezier curves. The point list is divided into consecutive sets of 4 points. Each set defines a single Bezier curve. If the points in a given set are designated $P_0..P_3$, then the defined Bezier curve goes from P_0 to P_3 using P_1 and P_2 as control points. The defined curve starts at P_0 and at P_0 is tangent to the line segment from P_0 to P_1. The curve ends at P_3 and at P_3 is tangent to the line segment from P_2 to P_3.

The derivation of the cubic parametric equations defining the curves is given in clause 4.

The individual Bezier curves, if there is more than one, are not implicitly connected. The Nth curve is connected to the (N-1)th curve only if the points P_{4N} and P_{4N-1} are identical.

References:

5.6.X PEL ARRAY

Parameters:

pel array identifier (I)
pel path (one of:0, 90, 180, 270) (E)
line progression (one of: 90, 270) (E)

pel spacing, pels per VDC (R)
line spacing, pels per VDC (R)
number of pels per line (I)
number of lines (I)
pel colour precision (I)
pel array (BS)

Description:

A compressed pel array image is defined as follows:

The pel array identifier is used as the tiling index when a tiled pel array is being defined. Otherwise the parameter has no significance.

The pel path parameter is the direction of progression of successive pels along a line relative to the VDC x-axis. The pel spacing and number of pels per line parameters together implicitly define the length and granularity for each line in the pel array, hence the pel array size in the pel path direction.

The line progression parameter is the direction of progression of successive of pel lines and is expressed as a direction relative to the pel path. Line spacing and the number of lines together implicitly define the size of the pel array in the direction of the line progression.

The compression method of the pel array is given by the current value of the PEL ARRAY COMPRESSION METHOD parameter. The sequence of pel colour specifiers is compressed according to the selected method and stored as a compressed binary data object in the metafile. The datatype is Bitstream.

The pel colour precision defines the colour precision of the colour specifiers in the pre-compressed or uncompressed datastream. Upon decompressing the Bitstream operand, these are the precisions of the binary data comprising the individual colour specifiers.

NOTE — T4 and T6 compression methods are not likely to give useful results if the pel colour precision is other than 1 and the colour selection mode is not indexed

References:
4.x

5.6.X BITONAL PEL ARRAY

Parameters:

pel array identifier (I)
pel path (one of:0, 90, 180, 270) (E)
line progression (one of: 90, 270) (E)
pel spacing, pels per VDC (R)
line spacing, pels per VDC (R)
number of pels per line (I)
number of lines (I)
pel background colour (CO)
pel foreground colour (CO)
pel array (BS)

Description:

A pel array image is defined as follows:

Elements

The pel array identifier is used as the tiling index when a tiled pel array is being defined. Otherwise the parameter has no significance.

The pel path parameter is the direction of progression of successive pels along a line relative to the VDC x-axis. The pel spacing and number of pels per line parameters together implicitly define the length and granularity for each line in the pel array, hence the pel array size in the pel path direction.

The line progression parameter is the direction of progression of successive of pel lines and is expressed as a direction relative to the pel path. Line spacing and the number of lines together implicitly define the size of the pel array in the direction of the line progression.

The compression method of the pel array is given by the current value of the PEL ARRAY COMPRESSION METHOD parameter. The sequence of pel colour specifiers is compressed according to the selected method and stored as a compressed binary data object in the metafile. The datatype is Bitstream.

The pel colour specifiers have two values, the indexes 0 and 1. Index 0 designates the pel background colour and index 1 designates the pel foreground colour. The precompressed or uncompressed colour specifiers, considered as a binary data stream, are represented at 1 bit per pel.

References:
4.x

5.6.X SYMBOL

Parameters:

point (P)
symbol index (IX)

Description:

The symbol corresponding to the symbol index parameter in the symbol library specified by the current SYMBOL LIBRARY INDEX is dimensioned according to SYMBOL HEIGHT, oriented according to SYMBOL ORIENTATION, and drawn at the specified position point. The symbol is displayed according to the current SYMBOL COLOUR.

Page 88

Subclause 5.7.19, CHARACTER SET INDEX, first line of description, after CHARACTER SET LIST add:

or GLYPH MAPPING

Page 90

Section 5.7.22, INTERIOR STYLE, in the Parameters section add to the end of enumerated list of styles:

interpolated

Page 95

Subclause 5.7.32, PATTERN TABLE, add the following at the end of the third paragraph of the description:

Note that COLOUR PRECISION only applies to direct colour (CD) values.

Page 96

Section 5.7.33, PATTERN SIZE, change the 3rd sentence of 3rd paragraph from:

This pattern box is divided into cells, ...

to:

When the selected pattern is a raster pattern — defined by the PATTERN TABLE element — this pattern box is divided into cells, ...

Page 97

Section 5.7.33, PATTERN SIZE, insert before the last paragraph:

When the pattern is a geometric pattern — defined by BEGIN/END GEOMETRIC PATTERN — the associated pattern extent rectangle — either default or defined by the GEOMETRIC PATTERN EXTENT — is mapped onto the pattern box parallelogram.

Page 97

Section 5.7.33, PATTERN SIZE, insert before the last paragraph:

When the pattern is a geometric pattern — defined by BEGIN/END GEOMETRIC PATTERN — the associated pattern extent rectangle — either default or defined by the GEOMETRIC PATTERN EXTENT — is mapped onto the pattern box parallelogram.

Page 98

Subclause 5.7, add the following attribute elements:

5.7.X LINE TYPE CONTINUATION

Parameters:

continuation mode (IX)

Description:

The behaviour of dashed line patterns at interior vertices of line elements is determined. Standardized values include:

1: unspecified — as in 8632 Version 1, any implementation dependent continuation is acceptable;
2: continue — the style is continued without interruption across vertices;

Elements

3: restart — the style is restarted at each vertex;
4: adaptive continue — the style is continued, but each vertex must be "inked".

The value 'adaptive continue' requires that each vertex contains a drawn portion of the pattern. This may require the pattern to be stretched or compressed.

Negative values are available for private methods, positive values are reserved for future standardization and registration. The latter may include some very specific requirements from application areas.

5.7.X LINE AND EDGE TYPE DEFINITION

Parameters:

linetype (IX)
dash unit selector (one of: VDC, proportion, fraction of display surface, abstract) (E)
dash repeat length (R)
list of dash elements (nI)

Description:

This element defines a line type or edge type and associates it with an index for future reference. The linetype parameter is the index of linetype being defined. It must be negative, to avoid conflict with standardized and registered values. The list of dash elements parameter comprises the definition to be associated with the index. The first element is a dash, second a space, etc. — the defined linetype is solid for I_1 units, gap for I_2 units, solid for I_3 units, and so on. There must be at least one element in the list of dash elements. If there is only one element in the list, a solid line is drawn. Each dash element must be non-negative. If an element is 0 for a drawn (versus gap) element of the dash element list then a dot is drawn.

The units of the dash repeat length parmeter are specified by the dash unit selector parameter. The value of 'abstract' indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion. The value 'proportion' means that the units are equal to the line width.

The dash repeat length defines the length of one complete cycle of the dash pattern, measured in the units of dash unit selector.

References:

5.7.X LINE TYPE INITIAL OFFSET

Parameters:

line pattern offset (R)

Description:

The line pattern offset is a real number between 0 and 1 which indicates how far into the current line pattern definition the drawing is actually started when a dashed line is begun.

5.7.X HATCH STYLE DEFINITION

Parameters:

hatch index (IX)
style indicator (one of: parallel, crosshatch) (E)
hatch space units selector (one of: VDC, fraction of display surface, abstract) (E)
hatch direction vectors (4VDC)
duty cycle length (R)
list of gap widths (nI)
list of line widths (nI)
list of line types (nIX)
list of line colours (nCI or nCD)
list of colour selection switches (one of: local colour, fill colour) (nE)

Description:

This element defines a hatch style and associates it with an index for future reference.

The hatch index parameter defines the index of hatch style by which the hatch style is subsequently referenced. The index must be negative, to avoid conflict with standardized and registered values.

The list of gap widths defines the gaps between the centers of the lines comprising the hatch.

Each line of the hatch definition has an associated series of line attributes: line type, line width, and line colour. The list of line widths defines the widths of the lines comprising the hatch. The list of line types defines the line types of the lines comprising the hatch. The list of line colours defines the line colours of the lines comprising the hatch, specified according to the current colour selection mode. The list of colour selection switches allows switching between the line colour specified and the fill colour for the definition of the colour of each line.

The centre of the first hatch line is aligned with the PATTERN REFERENCE POINT.

The hatch space units selector specifies the units of the duty cycle length.

The duty cycle length is measured perpendicular to the hatch lines. The sum of hatch line gaps in the hatch element list is normalized to this distance before presentation of the hatch on the view surface.

The hatch direction vectors specify the directions of the hatch lines. Only the first vector is significant if the hatch type is 'parallel'.

Hatch styles do not transform with the rotating and scaling transformations of CGM.

References:

5.7.X LINE CAP

Parameters:

line cap indicator (IX)
dash cap indicator (one of: off, on) (E)

Description:

The line cap and dash cap style is defined for subsequent line elements. The line cap indicator determines the appearance of open endpoints (as opposed to interior vertices) of line elements. The following values are assigned:

Elements

1: unspecified — as in 8632 version 1, any implementation dependent treatment is acceptable.
2: butt — the line is squared off at the endpoint, there is no projection beyond the endpoint.
3: round cap — a semicircular arc with diameter equal to the line width is drawn around the endpoint and filled in. The drawn line thus projects beyond the endpoint.
4: projecting square cap — the line is squared off at a distance equal to half the line width beyond the endpoint.
5: triangle — a cap is added to the line which is an equilateral triangle whose side equals the line width.

The dash cap indicator determines the appearance of the endpoints of individual dashs for subsequent dashed lines. When it is 'on' the endpoints of all the dashes have the style defined by the line cap indicator. When it is 'off' the endpoints of all dashes have the butt style, except for the open endpoints of the lines, which have the style defined by the line cap indicator.

The shape of line caps is not affected by transformation.

NOTE — Line cap values are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority When a line cap value has been approved by the ISO Subcommittee for Computer Graphics the line cap value will be assigned by the Registration Authority

References:

5.7.X LINE JOIN

Parameters:

line join indicator (IX)

Description:

The line join style is defined for subsequent line elements. The line join style defines the appearance of interior vertices of polyline elements and of compound line elements. The defined values are:

1: unspecified — as in 8632 version 1, any implementation dependent treatment is acceptable.
2: miter join — the outer edges of the two adjoining line segments are extended until they meet at a point.
3: round join — a circular arc with diameter equal to the line width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.
4: bevel join — the adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.7.X EDGE TYPE CONTINUATION

Parameters:

continuation mode (IX)

Description:

The behaviour of dashed edge patterns at the vertices of filled-area elements is determined. Standardized values include:

1: unspecified — as in 8632 Version 1, any implementation dependent continuation is acceptable;
2: continue — the style is continued without interruption across vertices;
3: restart — the style is restarted at each vertex;
4: adaptive continue — the style is continued, but each vertex must be "inked".

The value 'adaptive continue' requires that each vertex contains a drawn portion of the pattern. This may require the pattern to be stretched or compressed.

Negative values are available for private methods, positive values are reserved for future standardization and registration. The latter may include some very specific requirements from application areas.

5.7.X EDGE TYPE INITIAL OFFSET

Parameters:

edge pattern offset (R)

Description:

The edge pattern offset is a real number between 0 and 1 which indicates how far into the current edge pattern definition the drawing is actually started when a dashed edge is begun.

5.7.X EDGE CAP

Parameters:

edge cap indicator (IX)
dash cap indicator (one of: off, on) (E)

Description:

The edge cap indicator and dash cap indicator are defined for subsequent edge elements. The edge cap indicator determines the appearance of open endpoints of filled area edges (such as may result from a mixture of visible and invisible edge segments). The defined styles are:

1: unspecified — as in 8632 version 1, any implementation dependent treatment is acceptable.
2: butt — the edge is squared off at the endpoint, there is no projection beyond the endpoint.
3: round cap — a semicircular arc with diameter equal to the edge width is drawn around the endpoint and filled in. The drawn edge thus projects beyond the endpoint.
4: projecting square cap — the edge is squared off at a distance equal to half the edge width beyond the endpoint.
5: triangle — a cap is added to the edge which is an equilateral triangle whose side equals the edge width.

The shape of edge caps is not affected by transformation.

The dash cap indicator determines the appearance of the endpoints of each dash for subsequent dashed edges. When it is 'on' the endpoints of all the dashes have the style defined by the edge cap indicator. When it is 'off' the endpoints of all dashes have the butt style, except for the open endpoints of the edges, which have the style defined by the edge cap indicator.

Elements

NOTE — Edge cap values are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority When a edge cap value has been approved by the ISO Subcommittee for Computer Graphics, the edge cap value will be assigned by the Registration Authority

References:

5.7.X EDGE JOIN

Parameters:

edge join indicator (IX)

Description:

The edge join style is defined for subsequent filled elements. The edge join style defines the appearance of interior vertices of filled area elements. The defined values are:

1: unspecified — as in 8632 version 1, any implementation dependent treatment is acceptable.
2: miter join — the outer edges of the two adjoining edge segments are extended until they meet at a point.
3: round join — a circular arc with diameter equal to the edge width is drawn around the vertex between the adjoining segments and is filled in, producing a rounded corner.
4: bevel join — the adjoining edge segments are terminated with a butt cap, and the resulting triangular notch is filled in.

References:

5.7.X MITER LIMIT

Parameters:

miter limit (R)

Description:

Mitered corners can extend very far beyond the line vertex if the angle between the adjoining line segments is small. Miter length is defined to be the distance from the point at which the inner edges of the adjoining line segments meet to the point at which the outer edges meet. If miter length exceeds the 'miter limit' parameter, then the miter is truncated at the miter limit.

Miter limit is measured as a scale factor applied to the current line or edge width. Miter limit applies to line elements and edges of filled areas.

References:

5.7.X TEXT SCORE TYPE

Parameters:

list of pairs (score type, score indicator) (n[IX,E])

Description:

The following values are defined for score type:

1: right score (equivalent to underscore in left-to-right writing mode);
2: left score (equivalent to overscore in left-to-right writing mode);
3: through score (equivalent to strikeout in left-to-right writing mode);
4: kendot (emphasis similar to underscore for Kanji)

The score indicator may be either 'on' or 'off'. The value 'on' indicates that the corresponding score is used. The value 'off' indicates that the corresponding score type is not used.

Any combination of score types may be active simultaneously.

The text score may be changed in Text Open State (TOS).

References:

Page 98

Subclause 5.7, add the following to the attribute elements:

5.7.x RESTRICTED TEXT TYPE

Parameters:

index (IX)

Description:

The RESTRICTED TEXT constrains text strings to be within a parallelogram. This attribute selects one of a number of ways of applying the restriction to the text string. Assigned values of the restricted text method index are

1: basic;
2: boxed;
3: isotropic;
4: justified.

The effects of these values are described in clause 4.

References:

5.7.X GENERALIZED TEXT PATH MODE

Parameters:

Elements

mode (one of: off, non-tangential, axis-tangential) (E)

Description:

This element specifies which path the text string is to follow. If the mode is 'off' then the path specified by the TEXT PATH element ('right', 'left', 'up', or 'down') is used. If the mode is 'non-tangential' the characters are positioned along the path and oriented as per the character orientation vectors but the character orientation axes are not rotated — each character has the same orientation regardless of the path direction. If the mode is 'axis-tangential' the x-axis of the character orientation axes is tangent to the path at the character position — the orientation of each character depends upon the path direction at the character's placement point. In particular, the character orientation vectors are rotated together through the angle of the tangent to the path at the placement point.

This element affects the TEXT, RESTRICTED TEXT and APPEND TEXT primitives.

References:

5.7.X SYMBOL LIBRARY INDEX

Parameters:

symbol library index (IX)

Description:

The symbol library index is set to the value specified by the parameter. The symbol index selects a symbol library from the symbol library list defined in the Metafile Descriptor.

Legal values of the symbol library index parameter are positive integers.

5.7.X SYMBOL COLOUR

Parameters:

symbol colour specifier

if the colour selection mode is 'indexed',
symbol colour index (CI)

if the colour selection mode is 'direct',
symbol colour value (CO)

Description:

The symbol colour index or symbol colour value is set as specified by the parameter(s).

NOTE — Colour may be an aspect of a symbol's definition in the symbol library Annex D gives recommendations on how to handle SYMBOL COLOUR when the symbol itself contains colour

References:

5.7.X SYMBOL HEIGHT

Parameters:

symbol height (VDC)

Description:

The symbol height is set to the value specified by the parameter. Subsequent symbol elements are displayed with this symbol height. See 4.6 for a list of symbol elements.

The parameter represents the desired display height of the symbol in VDC units. It is measured along the symbol up vector. See 4.6 for a description of how symbols are sized and oriented for display.

Valid values of 'symbol height' are positive VDC.

References:

5.7.X SYMBOL ORIENTATION

Parameters:

x symbol up component (VDC)
y symbol up component (VDC)
x symbol base component (VDC)
y symbol base component (VDC)

Description:

The two vectors define the orientation and skew of the symbol in subsequent symbol elements. See 4.6 for a list of symbol elements. The ratio of the length of the base vector to the length of the up vector is used as a scaling factor to alter the design aspect ratio of the symbol, and the directions of the vectors determine rotation and skewing of the the symbol. See 4.6 for a description of how symbols are sized and oriented for display.

Valid values for the vectors include any which have non-zero length. and are not collinear.

References:

5.7.X GEOMETRIC PATTERN EXTENT

Parameters:

first point (P)
second point (P)

Description:

The first point and second point define two corners of a rectangular extent. The defined pattern extent rectangle is mapped to the pattern box parallelogram as described under the PATTERN SIZE element when a filled area element is displayed with a geometric pattern interior.

This element may occur only between BEGIN GEOMETRIC PATTERN and END GEOMETRIC PATTERN. If the element is not present then the pattern extent is the default defined in Clause 6.

Valid values for the two points are those which define a rectangle with positive area.

5.7.X INTERPOLATED INTERIOR DEFINITION

Parameters:

reference points (2P)
reference colours (2CO)
style (IX)

Description:

A solid but continuously graded colour interior is defined for filled area primitives. Conceptually a graded-colour infinite plane is defined in VDC space and this is "extruded" through the interior of filled area primitives to define the appearance of the interior.

The style parameter selects the way of defining the coloured plane. The following values are assigned:

1: circular
2: parallel

When the method is circular, then the first point is the centre of a circle and the second point is any point on the circumference. The first colour of the reference colours parameter applies at the centre and the second colour applies on the circumfrence. The colour at any interior point of the circle is the linear interpolant of the first and second colours along the straight line from the centre, through the interior point, to the circumfrence. The colours outside the circle are constant and equal to the second reference colour.

When the method is 'parallel', the coloured plane is defined as follows. The first colour is associated with the first point and the second colour with the second point. An infinite straight line — the reference line — is drawn through the two points. The reference line is composed of a finite interval — between the two reference points — and two semi-infinite rays — one starting at each reference point and going to infinity. The colour at any point on the reference line in the finite interval between the two reference points is the linear interpolant of the two reference colours at that point. The colour on each of the semi-infinite rays outside of the finite interval is constant and equal to the reference colour at the reference point. Colours are constant on lines perpendicular to the reference line and equal to the value at the intersection with the reference line.

Valid values of the reference points are any two distinct VDC points.

The linear interpolation is performed in the colour space of the colour model of the metafile.

NOTE — Styles are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a geometric pattern has been approved by the ISO Subcommittee on Computer Graphics, the pattern index value will be assigned by the Registration Authority.

5.7.X PEL ARRAY COMPRESSION METHOD

Parameters:

compression method (IX)

Description:

The compression method parameter specifies the compression format used to encode the image. The following values are defined:

1: T4;
2: T6;
3: LZW;
4: bitmap (uncompressed);
5: null background;
6: null foreground;

If the method is T4 the image is encoded according the one or two dimensional scheme defined in CCITT Recommendation T.4 (Group 3 facsimile). If the value is T6 the image is encoded according to the two dimensional scheme defined in CCITT Recommendation T.6 (Group 4 facsimile). Null background and null foreground indicate that all pels in the tile are known to be background or foreground respectively. In this case the pel array element has no encoded content — the Bitstream parameter is null.

References:

5.7.X PEL ARRAY REFERENCE POINT

Parameters:

reference point (P)

Description:

The reference point defines the position of subsequent PEL ARRAY or BITONAL PEL ARRAY elements. The first pel is placed at the reference point and subsequent pels are placed according to the parameters of the pel array element. If the pel path and line progression are thought of as vectors, the reference point is defined as point of origin for the two vectors.

References:
4.x

6 Defaults

Page 100

Clause 6: Add the following default specifications:

PEL ARRAY REFERENCE POINT	upper left-hand corner point of the default VDC extent
Protected region	default VDC Extent
PROTECTED REGION INDICATOR	off
COLOUR MODEL	1 (RGB)
PICTURE MAPPING	identity
COLOUR CALIBRATION	reference white value, D65
PEL ARRAY COMPRESSION METHOD	4 (bitmap)
GENERALIZED TEXT PATH MODE	1 (off)
RESTRICTED TEXT METHOD	1 (as 8632 version 1)
LINE TYPE CONTINUATION	1 (as 8632 version 1)
EDGE TYPE CONTINUATION	1 (as 8632 version 1)
LINE TYPE INITIAL OFFSET	0.0
EDGE TYPE INITIAL OFFSET	0.0
LINE CAP	1 (as 8632 version 1)
LINE JOIN	1 (as 8632 version 1)
EDGE CAP	1 (as 8632 version 1)
EDGE JOIN	1 (as 8632 version 1)
MITER LIMIT	1.0
SCORE TYPE	all text scores are 'off'
RESTRICTED TEXT TYPE	1
SYMBOL LIBRARY INDEX	n/a
SYMBOL LIBRARY LIST	n/a
INTERPOLATED INTERIOR DEFINITION	reference points — VDC extent reference colours — device-dependent background colour if COLOUR SELECTION MODE is 'direct',

GEOMETRIC PATTERN EXTENT	0 if COLOUR SELECTION MODE is 'indexed' default VDC extent

7 Annex D

Page 192

Subclause D.4.6, add the following new paragraphs at the end:

SYMBOL COLOUR

It is implementation dependent how a CGM interpreter applies SYMBOL COLOUR if a symbol is selected whose definition includes colour information.

Tiled Pel Array

If the number of tiles present does not match the count specified by the BEGIN TILE PEL ARRAY parameter, it is recommended that the missing tiles be treated as encoded as "null background".

ISO 8632/Am.3

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer of Picture Description Information

Part 2

Character Encoding

5 Method of Encoding Opcodes

Page 12

Clause 5, Table 1, add the following opcodes

Table 1 — Opcodes for metafile elements.

Opcode	7-Bit coding		8-Bit coding	
BEGIN GEOMETRIC PATTERN opcode	3/0	2/9	03/3	02/9
END GEOMETRIC PATTERN opcode	3/0	2/10	03/0	02/10
BEGIN COMPOUND PATH opcode	3/0	2/11	03/0	02/11
END COMPOUND PATH opcode	3/0	2/12	03/0	02/12
BEGIN PROTECTED REGION opcode	3/0	2/13	03/0	02/13
END PROTECTED REGION opcode	3/0	2/14	03/0	02/14
BEGIN TILED PEL ARRAY opcode	3/0	2/15	03/0	02/14
END TILED PEL ARRAY opcode	3/0	3/0	03/0	03/0
COLOUR MODEL opcode	3/1	3/3	03/1	03/3
COLOUR CALIBRATION opcode	3/1	3/4	03/1	03/4
FONT PROPERTIES opcode	3/1	3/5	03/1	03/5
GLYPH MAPPING opcode	3/1	3/6	03/1	03/6
SYMBOL LIBRARY LIST opcode	3/1	3/7	03/1	03/7
PICTURE MAPPING opcode	3/2	2/15	03/2	02/15
PROTECTED REGION INDICATOR opcode	3/3	2/12	03/3	02/12
DELETE PROTECTED REGION opcode	3/3	2/13	03/3	02/13
HYPERBOLIC ARC opcode	3/4	2/10	03/4	02/10
PARABOLIC ARC opcode	3/4	2/11	03/4	02/11
NON-UNIFORM B-SPLINE opcode	3/4	2/12	03/4	02/12
POLYBEZIER opcode	3/4	2/13	03/4	02/13
SYMBOL opcode	3/4	2/14	03/4	02/14
BITONAL PEL ARRAY opcode	3/4	2/15	03/4	02/15
PEL ARRAY opcode	3/4	3/0	03/4	03/0
LINE AND EDGE TYPE DEFINITION opcode	3/5	2/8	03/5	02/8
HATCH STYLE DEFINITION opcode	3/5	2/9	03/5	02/9
LINE CAP opcode	3/5	2/10	03/5	02/10
LINE JOIN opcode	3/5	2/11	03/5	02/11
LINE MITER LIMIT opcode	3/5	2/12	03/5	02/12
EDGE CAP opcode	3/5	2/13	03/5	02/13
EDGE JOIN opcode	3/5	2/14	03/5	02/14
EDGE MITER LIMIT opcode	3/5	2/15	03/5	02/15
TEXT SCORE TYPE opcode	3/6	2/13	03/6	02/13
RESTRICTED TEXT TYPE opcode	3/6	2/14	03/6	02/14
GENERALIZED TEXT PATH MODE opcode	3/6	2/15	03/6	02/15
LINE TYPE CONTINUATION opcode	3/6	3/3	03/6	03/3
LINE TYPE INITIAL OFFSET opcode	3/6	3/4	03/6	03/4
EDGE TYPE CONTINUATION opcode	3/6	3/5	03/6	03/5
EDGE TYPE INITIAL OFFSET opcode	3/6	3/6	03/6	03/6
GEOMETRIC PATTERN EXTENT opcode	3/6	3/7	03/6	03/7
INTERPOLATED INTERIOR DEFINITION opcode	3/6	3/8	03/6	03/8
SYMBOL LIBRARY INDEX opcode	3/6	3/9	03/6	03/9
SYMBOL COLOUR opcode	3/6	3/10	03/6	03/10
SYMBOL HEIGHT opcode	3/6	3/11	03/6	03/11
SYMBOL ORIENTATION opcode	3/6	3/12	03/6	03/12
PEL ARRAY REFERENCE POINT opcode	3/6	3/13	03/6	03/13
PEL ARRAY COMPRESSION METHOD opcode	3/6	3/14	03/6	03/14

6 Method of encoding parmameters

***To be completed.

8 Representation of each element

Page 91

Subclause 8.1, add the following Delimiter element representations:

8.1.x BEGIN GEOMETRIC PATTERN

<BEGIN-GEOMETRIC-PATTERN-opcode: 3/0 2/6>
<integer: pattern-table-index>

<integer: pattern-table-index> = <positive-integer>

8.1.x END GEOMETRIC PATTERN

<END-GEOMETRIC-PATTERN-opcode: 3/0 2/7>

8.1.x BEGIN COMPOUND PATH

<BEGIN-COMPOUND-PATH-opcode: 3/0 2/8>
<enumerated: path-type>

<enumerated: path-type> = <integer: 0> {text path}
| <integer: 1> {compound line}

8.1.x END COMPOUND PATH

<END-COMPOUND-PATH-opcode: 3/0 2/9>

8.1.x BEGIN PROTECTED REGION

<BEGIN-PROTECTED-REGION-opcode: 3/0 2/10>
<index: region-index>
<enumerated: region-type>

<index: region-index> = <positive integer>

<enumerated: region-type> = <integer: 0> {clip}
| <integer: 1> {shield}

8.1.x END PROTECTED REGION

<END-PROTECTED-REGION-opcode: 3/0 2/11>

8.1.x BEGIN TILED PEL ARRAY

<BEGIN-TILED-PEL-ARRAY-opcode: 3/0 2/12>
<integer: tiled-pel-array-dimension-x>
<integer: tiled-pel-array-dimension-y>
<integer: tiling-offset-x>
<integer: tiling-offset-y>
<integer: image-size-x>
<integer: image-size-y>

Representation of each element

<integer: tiled-pel-array-dimension-x>	=	<positive integer>
<integer: tiled-pel-array-dimension-y>	=	<positive integer>
<integer: tiling-offset-x>	=	<non-negative integer>
<integer: tiling-offset-y>	=	<non-negative integer>
<integer: image-size-x>	=	<positive integer>
<integer: image-size-y>	=	<positive integer>

8.1.x END TILED PEL ARRAY

<END-TILED-PEL-ARRAY-opcode: 3/0 2/13>

Page 33

Subclause 8.2: Add the following Metafile Descriptor element representations:

8.2.x COLOUR MODEL

<COLOUR-MODEL-opcode: 3/1 3/3>
<enumerated: colour-model>

<enumerated: colour-model>	=	<integer: 0> {RGB}
	|	<integer: 1> {CIELAB}
	|	<integer: 2> {CMYK}

8.2.x COLOUR CALIBRATION

<COLOUR-CALIBRATION-opcode: 3/1 3/4>
<real: reference-white-Xn>
<real: reference-white-Yn>
<real: reference-white-Zn>
<calibration-data-matrix>

<calibration-data-matrix>	=	<real: Xr> <real: Xg> <real: Xb>
		<real: Yr> <real: Yg> <real: Yb>
		<real: Zr> <real: Zg> <real: Zb>

8.2.x FONT PROPERTIES

<FONT-PROPERTIES-opcode: 3/1 3/5>
<index: font-index>
<string: typeface-name>
<string: font-family>
<octet: typeface-design-group-general-class>
<octet: typeface-design-group-subclass>
<octet: typeface-design-group-specific-group>
<index: posture>
<real: posture-angle>
<index: weight>
<index: proportional width>
<string: structure>

<index: font-index>	=	<positive integer>
<string: typeface-name>	=	<string: name of typeface>

<string: font-family> = <string: name of font family>

<octet: typeface-design-group-general-class> = <integer: 0..255>
<octet: typeface-design-group-subclass> = <integer: 0..255>
<octet: typeface-design-group-specific-group> = <integer: 0..255>

<index: posture> = <integer: 0> {not applicable}
| <integer: 1> {upright}
| <integer: 2> {oblique}
| <integer: 3> {back slanted oblique}
| <integer: 4> {italic}
| <integer: 5> {back slanted italic}
| <integer: 6> {other}

<real: posture-angle> = <real: -360..+360>

<index: weight> = <integer: 0> {not applicable}
| <integer: 1> {ultra light}
| <integer: 2> {extra light}
| <integer: 3> {light}
| <integer: 4> {semi light}
| <integer: 5> {medium}
| <integer: 6> {semi bold}
| <integer: 7> {bold}
| <integer: 8> {extra bold}
| <integer: 9> {ultra bold}

<index: proportional width> = <integer: 0> {not applicable}
| <integer: 1> {ultra condensed}
| <integer: 2> {extra condensed}
| <integer: 3> {condensed}
| <integer: 4> {semi condensed}
| <integer: 5> {medium}
| <integer: 6> {semi expanded}
| <integer: 7> {expanded}
| <integer: 8> {extra expanded}
| <integer: 9> {ultra expanded}

8.2.x GLYPH MAPPING

<GLYPH-MAPPING-opcode: 3/1 3/6>
<index: character-set-index>
<string: basis-set>
<integer: octets-per-code>
<code-glyph-name-pair>+

<index: character-set-index> = <positive-integer>

<string: basis-set> = <string: end-of-escape-sequence-to-designate-set>

<integer: octets-per-code> = <positive-integer>

<code-glyph-name-pair>+ = <character-code> <afii-4-byte-glyph-identifier>

<character-code> = <octet>(octets-per-code)

Representation of each element

<afii-4-byte-identifier> = <octet>(4)

8.2.x SYMBOL LIBRARY LIST

<SYMBOL-LIBRARY-LIST-opcode: 3/1 3/7>
<symbol-library-name>+

<symbol-library-name> = <string: name-of-symbol-library>

Page 37

Subclause 8.3, add the following Picture Descriptor element representations:

8.3.x PICTURE MAPPING

<PICTURE-MAPPING-opcode: 3/2 2/7>
<picture-mapping-matrix>

<picture-mapping-matrix> = <real: a11> <real: a12> <real: a13>
| <real: a21> <real: a22> <real: a23>
| <real: a31> <real: a32> <real: a33>

Page 39

Subclause 8.4, Add the following Control element representations:

8.4.x PROTECTED REGION INDICATOR

<PROTECTED-REGION-INDICATOR-opcode: 3/3 2/12>
<index: region-index>
<enumerated: region-indicator>

<index: region-index> = <positive integer>

<enumerated: region-indicator> = <integer: 0> {off}
| <integer: 1> {on}

8.4.x DELETE PROTECTED REGION

<DELETE-PROTECTED-REGION-opcode: 3/3 2/13>
<index: region-index>

<index: region-index> = <positive integer>

Page 41

Subclause 8.5, Add the following Graphical Primitive element representations:

8.5.x HYPERBOLIC ARC

<HYPERBOLIC-ARC-opcode: 3/4 2/10>

<point: centre-point>
<point: transverse-radius-endpoint>
<point: conjugate-radius-endpoint>
<VDC: DX_start>
<VDC: DY_start>
<VDC: DX_end>
<VDC: DY_end>

8.5.x PARABOLIC ARC

<PARABOLIC-ARC-opcode: 3/4 2/11>
<point: tangent-intersection-point>
<point: start-point>
<point: end-point>

8.5.x NON-UNIFORM B-SPLINE

<NON-UNIFORM-B-SPLINE-opcode: 3/4 2/12>
<integer: spline_order>
<list-of-knots>
<enumerated: rationality>
<control-points>
<real: parameter-start-value>
<real: parameter-end-value>

<integer: spline_order>	=	<positive integer>
<list-of-knots>	=	<real: knot>+
<enumerated: rationality>	=	<integer: 0> {rational}
	|	<integer: 1> {non-rational}
<control-points>	=	<point: control-point>+
	|	<homogeneous-point: control-point>+

8.5.x POLYBEZIER

<POLYBEZIER-opcode: 3/4 2/13>
<control-points>

<control-points>	=	<point>(4N)

8.5.x SYMBOL

<SYMBOL-opcode: 3/4 2/14>
<point: symbol-position>
<index: symbol-index>

<index: symbol-index>	=	<positive integer>

8.5.x BITONAL PEL ARRAY

<BITONAL-PEL-ARRAY-opcode: 3/4 2/15>
<integer: pel-array-identifier>
<enumerated: pel-path>
<enumerated: line-progression>

Representation of each element

<real: pel-spacing>
<real: line-spacing>
<integer: number-of-pels-per-line>
<integer: number-of-lines>
<colour specifier: pel background colour>
<colour specifier: pel foreground colour>
<bitstream: pel-array>

<integer: pel-array-identifier>	=	<positive integer>
<enumerated: P-pel-path>	=	<integer: 0> {0-degrees}
	|	<integer: 1> {90-degrees}
	|	<integer: 2> {180-degrees}
	|	<integer: 3> {270-degrees}
<enumerated: L-line-progression>	=	<integer: 0> {90-degrees}
	|	<integer: 1> {270-degrees}
<integer: number-of-pels-per-line>	=	<positive integer>
<integer: number-of-lines>	=	<positive integer>
<integer: pel-colour-precision>	=	<positive integer>

8.5.x PEL ARRAY

<PEL-ARRAY-opcode: 3/4 3/0>
<integer: pel-array-identifier>
<enumerated: pel-path>
<enumerated: line-progression>
<real: pel-spacing>
<real: line-spacing>
<integer: number-of-pels-per-line>
<integer: number-of-lines>
<integer: pel-colour-precision>
<bitstream: pel-array>

<integer: pel-array-identifier>	=	<positive integer>
<enumerated: P-pel-path>	=	<integer: 0> {0-degrees}
	|	<integer: 1> {90-degrees}
	|	<integer: 2> {180-degrees}
	|	<integer: 3> {270-degrees}
<enumerated: L-line-progression>	=	<integer: 0> {90-degrees}
	|	<integer: 1> {270-degrees}
<integer: number-of-pels-per-line>	=	<positive integer>
<integer: number-of-lines>	=	<positive integer>
<integer: pel-colour-precision>	=	<positive integer>

Page 46

Subclause 8.6, Add the following Primitive Attribute element representations:

8.6.x LINE AND EDGE TYPE DEFINITION

<LINE-AND-EDGE-TYPE-DEFINITION-opcode: 3/5 2/8>
<index: line-type>
<enumerated: dash-unit-selector>
<real: dash-repeat-length>
<integer: list-of-dash-elements>

<index: line-type>	=	<negative-integer>
<enumerated: dash-unit-selector>	=	<integer: 0> {VDC}
	|	<integer: 1> {proportion}
	|	<integer: 2> {fraction of display surface}
	|	<integer: 3> {abstract}
<integer: list-of-dash-elements>	=	<positive-integer>+

8.6.x HATCH STYLE DEFINITION

<HATCH-STYLE-DEFINITION-opcode: 3/5 2/9>
<integer: hatch-index>
<enumerated: style-indicator>
<enumerated: hatch-space-units-selector>
<VDC: DX-first-vector>
<VDC: DY-first-vector>
<VDC: DX-second-vector>
<VDC: DY-second-vector>
<real: duty-cycle-length>
<list-of-gap-widths>
<list-of-line-widths>
<list-of-line-types>
<list-of-line-colours>
<list-of-colour-selection-switches>

<integer: hatch-index>	=	<negative-integer>
<enumerated: style-indicator>	=	<integer: 0> {parallel}
	|	<integer: 1> {cross-hatch}
<enumerated: hatch-space-units-selector>	=	<integer: 0> {VDC}
	|	<integer: 1> {fraction of display surface}
	|	<integer: 2> {abstract}
<integer: list-of-hatch-elements>	=	<positive-integer>+
<real: duty-cycle-length>	=	<positive real>
<list-of-gap-widths>	=	<positive integer>+
<list-of-line-widths>	=	<positive integer>+

Representation of each element

<list-of-line-types>	=	<positive integer>+
<list-of-line-colours>	=	<colour specifier>+
<list-of-colour-selection-switches>	=	<enumerated: switch-value>+
<enumerated: switch-value>	=	<integer: 0> {local colour}
	|	<integer: 1> {fill colour}

8.6.x LINE CAP

<LINE-CAP-opcode: 3/5 2/10>
<index: line-cap-indicator>
<enumerated: dash-cap-indicator>

<index: line-cap-indicator>	=	<integer: 1> {unspecified}
	|	<integer: 2> {butt}
	|	<integer: 3> {round}
	|	<integer: 4> {projecting square}
	|	<integer: 5> {triangle}
<enumerated: dash-cap-indicator>	=	<integer: 0> {off}
	|	<integer: 1> {on}

8.6.x LINE JOIN

<LINE-JOIN-opcode: 3/5 2/11>
<index: line-join-indicator>

<index: line-join-indicator>	=	<integer: 1> {unspecified}
	|	<integer: 2> {miter}
	|	<integer: 3> {round}
	|	<integer: 4> {bevel}

8.6.x LINE MITER LIMIT

<LINE-MITER-LIMIT-opcode: 3/5 2/12>
<real: line-miter-limit>

<real: line-miter-limit>	=	<non-negative-real>

8.6.x EDGE CAP

<EDGE-CAP-opcode: 3/5 2/13>
<index: edge-cap-indicator>
<enumerated: dash-cap-indicator>

<index: edge-cap-indicator>	=	<integer: 1> {unspecified}
	|	<integer: 2> {butt}
	|	<integer: 3> {round}
	|	<integer: 4> {projecting square}
	|	<integer: 5> {triangle}
<enumerated: dash-cap-indicator>	=	<integer: 0> {off}
	|	<integer: 1> {on}

8.6.x EDGE JOIN

<EDGE-JOIN-opcode: 3/5 2/14>
<index: edge-join-indicator>

<index: edge-join-indicator>	=	<integer: 1> {unspecified}
	|	<integer: 2> {miter}
	|	<integer: 3> {round}
	|	<integer: 4> {bevel}

8.6.x EDGE MITER LIMIT

<EDGE-MITER-LIMIT-opcode: 3/5 2/15>
<real: edge-miter-limit>

<real: edge-miter-limit>	=	<non-negative-real>

8.6.x TEXT SCORE TYPE

<TEXT-SCORE-TYPE-opcode: 3/6 2/13>
<type-and-indicator-pair>+

<type-and-indicator-pair>	=	<score type> <score indicator>
<index: score-type>	=	<integer: 1> {right score}
	|	<integer: 2> {left score}
	|	<integer: 3> {through score}
	|	<integer: 4> {kendot}
<enumerated: score indicator>	=	<integer: 0> {off}
	|	<integer: 1> {on}

8.6.x RESTRICTED TEXT TYPE

<RESTRICTED-TEXT-TYPE-opcode: 3/6 2/14>
<index: restricted-text-type>

<index: restricted-text-type>	=	<integer: 1> {basic}
	|	<integer: 2> {boxed}
	|	<integer: 3> {isotropic}
	|	<integer: 4> {justified}

8.6.x GENERALIZED TEXT PATH MODE

<GENERALIZED-TEXT-PATH-MODE-opcode: 3/6 2/15>
<enumerated: path-text-mode>

<enumerated: path-text-mode>	=	<integer: 0> {off}
	|	<integer: 1> {non-tangential}
	|	<integer: 2> {axis-tangential}

8.6.x LINE TYPE CONTINUATION

<LINE-TYPE-CONTINUATION-opcode: 3/6 3/3>
<index: continuation-mode>

<index: continuation-mode>	=	<integer: 1> {unspecified}

Representation of each element

	|	<integer: 2> {continue}
	|	<integer: 3> {restart}
	|	<integer: 4> {adaptive continue}

8.6.x LINE TYPE INITIAL OFFSET

<LINE-TYPE-INITIAL-OFFSET-opcode: 3/6 3/4>
<real: line-pattern-offset>

8.6.x EDGE TYPE CONTINUATION

<EDGE-TYPE-CONTINUATION-opcode: 3/6 3/5>
<index: continuation-mode>

<index: continuation-mode>	=	<integer: 1> {unspecified}
	|	<integer: 2> {continue}
	|	<integer: 3> {restart}
	|	<integer: 4> {adaptive continue}

8.6.x EDGE TYPE INITIAL OFFSET

<EDGE-TYPE-INITIAL-OFFSET-opcode: 3/6 3/6>
<real: edge-pattern-offset>

8.6.x GEOMETRIC PATTERN EXTENT

<GEOMETRIC-PATTERN-EXTENT-opcode: 3/6 3/7>
<point: first-point>
<point: second-point>

8.6.x INTERPOLATED INTERIOR DEFINITION

<INTERPOLATED-INTERIOR-DEFINITION-opcode: 3/6 3/8>
<point: first-reference-point>
<point: second-reference-point>
<colour-specifier: first-reference-colour>
<colour-specifier: second-reference-colour>
<index: method>

<colour-specifier>	=	<integer: colour-index> {if COLOUR SELECTION MODE is indexed}
	|	<RGB> {if COLOUR SELECTION MODE is direct}
<index: method>	=	<integer: 1> {circular}
	|	<integer: 2> {parallel}

8.6.x SYMBOL LIBRARY INDEX

<SYMBOL-LIBRARY-INDEX-opcode: 3/6 3/9>
<index: symbol-library-index>

8.6.x SYMBOL COLOUR

<SYMBOL-COLOUR-opcode: 3/6 3/10>
<colour-specifier>

<colour-specifier>	=	<integer: colour-index> {if COLOUR SELECTION MODE is indexed}

| <RGB> {if COLOUR SELECTION MODE is direct}

8.6.x SYMBOL HEIGHT

<SYMBOL-HEIGHT-opcode: 3/6 3/11>
<VDC: symbol-height>

8.6.x SYMBOL ORIENTATION

<SYMBOL-ORIENTATION-opcode: 3/6 3/12>
<VDC: x-component of up vector>
<VDC: y-component of up vector>
<VDC: x-component of base vector>
<VDC: y-component of base vector>

8.6.x PEL ARRAY REFERENCE POINT

<PEL-ARRAY-REFERENCE-POINT-opcode: 3/6 3/13>
<point: reference-point>

8.6.x PEL ARRAY COMPRESSION METHOD

<PEL-ARRAY-COMPRESSION-METHOD-opcode: 3/6 3/14>
<index: compression-method>

<index: compression-method>	=	<integer:1> {T4}
	|	<integer:1> {T6}
	|	<integer:2> {LZW}
	|	<integer:3> {bitmap}
	|	<integer:4> {null background}
	|	<integer:5> {null foreground}

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer

of Picture Description Information

Part 3

Binary Encoding

Amendment 3

Draft Document 1.4

April 13, 1990

Page 16

Add the following at the end of table 1:

BS	UI at fixed precision (16-bit) {see note 14}	BBS {=2}	BSR {see note 14}

Page 18

Add the following note at the end:

14) The Bit Stream data type consists of a stream of binary digits (bits) packed in 16-bit unsigned integers. The stream is constrained to end on a word boundary. The range for parameter type BS is not applicable.

Page 20

Subclause 7.2: Add the following to Table 3:

BEGIN COMPOUND PATH	6	E	BE	{0,1}	1
END COMPOUND PATH	7	n/a	0	n/a	n/a
BEGIN PROTECTED REGION	8	IX,E	BIX,BE	IXR{0,1}	see below
END PROTECTED REGION	9	n/a	0	n/a	n/a
BEGIN TILED PEL ARRAY	10	6I	6BI+	+IR	n/a
END TILED PEL ARRAY	11	n/a	0	n/a	n/a
BEGIN GEOMETRIC PATTERN	12	IX	BIX	IXR	n/a
END GEOMETRIC PATTERN	13	n/a	0	n/a	n/a

Add the following notes (on Table 4):

nn BEGIN COMPOUND PATH: has 1 parameter:

P1: (enumerated) path type: valid values are:

0 text path
1 compound line

nn END COMPOUND PATH: has no parameters.

nn BEGIN PROTECTED REGION: has 2 parameters:

P1: (index) region index.
P2: (enumerated) region type: valid values are

0 clip
1 shield

nn END PROTECTED REGION: has no parameters.

nn BEGIN TILED PEL ARRAY: has 6 parameters:

P1: (integer) tiled pel array dimension in the pel path direction.
P2: (integer) tiled pel array dimension in the line progression direction.
P3: (integer) tiling offset in the pel path direction.

P4: (integer) tiling offset in the line progression direction.
P5: (integer) image size in the pel path direction.
P6: (integer) image size in the line progression direction.

nn END TILED PEL ARRAY: has no parameters.

nn BEGIN GEOMETRIC PATTERN: has 1 parameter:

P1: (index) pattern table index.

nn END GEOMETRIC PATTERN: has no parameters.

Page 21

Subclause 7.3: Add the following to Table 4:

COLOUR MODEL	19	IX	BIX	IXR	1
FONT PROPERTIES	20	IX,2S,	BIX,BS	++IXR,SR	0,null
		3IX,	3BIX	IXR	see below
		IX	BIX	IXR	0
		R,	BR	RR	0.00
		2IX,	2BIX	IXR	0,0
		S	BS	SR	null
GLYPH MAPPING	21	IX,S	BIX+,BS	+IXR,SR	1,null
		I	BI+	+IR	2
		mnOC	mnBOC	OCR	n/a
		4nOC	nBOC	OCR	n/a
COLOUR CALIBRATION	22	12R	12BR	RR	see below
SYMBOL LIBRARY LIST	23	nS	nBS	SR	n/a
PICTURE MAPPING	24	9R	9BR	RR	see below

Add the following notes (on Table 4):

nn COLOUR MODEL: has 1 parameter:

P1: (index) colour model: the following values are standardized:

0 RGB
1 CIELAB
2 CMYK

nn FONT PROPERTIES: has 11 parameters:

P1: (index) font index
P2: (string) typeface name
P3: (string) font family
P4: (index) typeface design group general class
P5: (index) typeface design group subclass
P6: (index) typeface design group specific group
P7: (index) posture: the following values are standardized:

0 not applicable
1 upright
2 oblique - upright design slanted in the nominal escapement direction with no design change

3	back-slanted oblique - udpright design slanted in opposite direction of nominal escapement direction with no design change
4	italic - slanted in nominal escapement direction with change in design
5	back-slanted italic - slanted in opposite direction of nominal escapement with change in design
6	other

P8: (real) posture angle
P9: (index) weight: the following values are standardized:

0	not applicable
1	ultra light
2	extra light
3	light
4	semi light
5	medium
6	semi bold
7	bold
8	extra bold
9	ultra bold

P10: (index) proportionate width: the following values are standardized:

0	not applicable
1	ultra condensed
2	extra condensed
3	condensed
4	semi condensed
5	medium
6	semi expanded
7	expanded
8	extra expanded
9	ultra expanded

P11: (string) structure

nn GLYPH MAPPING: has 4 parameters:

P1: (index) character set index
P2: (string) basic set
P3: (integer) octets per code
P4: (octet pair array) character code of length specified by the octets per code parameter, followed by the 4 octet glyph name associated with that code.

nn COLOUR CALIBRATION: has 13 parameters

P1: (real) reference white value X component
P2: (real) reference white value Y component
P3: (real) reference white value Z component
P4-P13: (real) 3X3 matrix of calibration data: row 1 of the

matrix consists of Xred, Xgreen, Xblues values; row 2 is Yred, Ygreen, Yblue; and row 3 is Zred, Zgreen, Zblue.

nn SYMBOL LIBRARY LIST: has a variable parameters

P1-Pn: array of symbol library names (strings), the first name in the list is assigned to index 1, the second to index 2, etc.

nn PICTURE MAPPING: has 9 parameters:

P1-P9: picture mapping transformation matrix: consists of 9 real values that define a 3x3 matrix that produces a transformation to be applied to all picture coordinates

Page 26

Add the following to table 6:

PROTECTED REGION INDICATOR	13	IX	BIX	IXR	n/a
		E	BE+	{0,1}	0
DELETE PROTECTED REGION	14	IX	BIX	IXR	1

Add the following notes (on table 6):

nn PROTECTED REGION INDICATOR: has 2 parameters:

P1: (index) region index
P1: (enumerated) region indicator: valid values are:

0 off
1 on

nn DELETE PROTECTED REGION: has 1 parameter:

P1: (index) region index

Page 28

Subclause 7.6: Add the following to Table 7:

HYPERBOLIC ARC	22	3P,4VDC	3BP,4BVDC	VDCR	n/a
PARABOLIC ARC	23	3P	3BP	VDCR	n/a
NON-UNIFORM B-SPLINE	24	I	BI+	+IR	n/a
		nR	nBR	RR	n/a
		E	BE	{0,1}	0
		mP,2R	mBP,BR	VDCR,RR	n/a
POLYBEZIER	25	4nP	4nBP	VDCR	n/a
PEL ARRAY	26	I	BI+	+IR	n/a
		2E	2BE+	{0,..3}, {0,1}	
		2R	2BR+	+RR	
		2I	2BI+	+IR	
		BS	BBS	BSR	
BITONAL PEL ARRAY	27	I	BI+	+IR	n/a
		2E	2BE+	{0,..3},	

SYMBOL	28	2R 2I BS P IX	2BR+ 2BI+ BBS BP BIX+	{0,1} +RR +IR BSR VDCR ++IXR	n/a

Add the following notes (on Table 7):

nn HYPERBOLIC ARC: has 7 parameters:

P1: (point) center point
P2: (point) tranverse radius end point
P3: (point) conjugate radius end point
P4: (vdc) start vector x component
P5: (vdc) start vector y component
P6: (vdc) end vector x component
P7: (vdc) end vector x component

nn PARABOLIC ARC: has 3 parameters:

P1: (point) tangent intersection point
P2: (point) start point
P3: (point) end point

nn NON-UNIFORM B-SPLINE: has a variable parameter list:

P1: (integer) spline order
P2-Pn: (real) list of knots
P(n+1): (enumerated) rationality: valid values are:

0 rational
1 non-rational

P(n+2)-P(m): (points) array of control points
P(m+1): (real) parameter start value
P(m+2): (real) parameter end value

nn POLYBEZIER: has a variable parameter list:

P1-P4n: (point) list of n 4 point sequences: each sequence defines a single bezier curve

nn PEL ARRAY: has 8 parameters:

P1: (integer) pel array identifier
P2: (enumerated) pel path: valid values are:

0 0 degrees
1 90 degrees
2 180 degrees
3 270 degrees

P3: (enumerated) line progression: valid values are:

0 90 degrees
1 270 degrees

P4: (real) pel spacing: specified in pels per VDC
P5: (real) line spacing: specified in pels per VDC
P6: (integer) number of pels per line
P7: (integer) number of lines
P8: (binary stream) pel array

nn BITONAL PEL ARRAY: has 8 parameters:

P1: (integer) pel array identifier
P2: (enumerated) pel path: valid values are:

0 0 degrees
1 90 degrees
2 180 degrees
3 270 degrees

P3: (enumerated) line progression: valid values are:

0 90 degrees
1 270 degrees

P4: (real) pel spacing: specified in pels per VDC
P5: (real) line spacing: specified in pels per VDC
P6: (integer) number of pels per line
P7: (integer) number of lines
P8: (binary stream) pel array

nn SYMBOL: has 2 parameters:

P1: (point) symbol position point
P2: (index) symbol index

Page 32

Subclause 7.7: Add the following to Table 8:

LINE TYPE CONTINUATION	36	IX	BIX	IXR	0
LINE AND EDGE TYPE DEFINITION	37	IX	BIX+	++IXR	
		E	BE	{0,..2)	
		R	BR	RR	
		nI	nBI	IR	
LINE TYPE INITIAL OFFSET	38	R	BR	RR	0.0
HATCH STYLE DEFINITION	39	IX	BIX	IXR	see below
		2E	2BE	{0,1}, {0,1,2}	0,0
		4VDC	4BVDC	VDCR	0,1,1,0
		R	BR	RR	see below
		2nI	2nBI	IR	n/a
		nIX	nBIX	IXR	n/a
		nCO	nBCO	COR	n/a

		nE	nBE	{0,1}	0
LINE CAP	40	IX	BIX	IXR	1
		E	BE	{0,1}	0
LINE JOIN	41	IX	BIX	IXR	1
EDGE TYPE CONTINUATION	42	IX	BIX	IXR	1
EDGE TYPE INTIAL OFFSET	43	R	BR	RR	0.00
EDGE CAP	44	IX	BIX	IXR	1
		E	BE	{0,1}	0
EDGE JOIN	45	IX	BIX	IXR	1
MITER LIMIT	46	R	BR	RR	1.00
TEXT SCORE TYPE	47	nIX,nE	nBIX,nBE	IXR {1,..4}	1,1
RESTRICTED TEXT TYPE	48	IX	BIX	IXR	1
GENERALIZED TEXT PATH MODE	49	E	BE	{0,1,2}	0
SYMBOL LIBRARY INDEX	50	IX	BIX	IXR	1
SYMBOL COLOUR	51	CO	BCO	COR	see below
SYMBOL HEIGHT	52	VDC	BVDC+	++VDCR	see below
SYMBOL ORIENTATION	53	4VDC	4BVDC	VDCR	0,1,1,0
GEOMETRIC PATTERN EXTENT	54	2P	2BP	VDCR	see below
INTERPOLATED INTERIOR DEFINITION	55	2P	2BP	VDCR	see below
		2CO	2BCO	COR	see below
		IX	BIX	IXR	1
PEL ARRAY COMPRESSION METHOD	56	IX	BIX	IXR	5
PEL ARRAY REFERENCE POINT	57	P	BP	VDCR	n/a

Add the following notes (on Table 8):

nn LINE TYPE CONTINUATION: has 1 parameter:

P1: (index) continuation mode: the following values are standardized:

1 unspecified
2 continue
3 restart
4 adaptive continue
negative values are available for private use

nn LINE AND EDGE TYPE DEFINITION: has a variable parameter list:

P1: (index) line type
P2: (enumerated) dash unit selector: valid values are:

0 VDC
1 proportion
2 fraction of display surface
3 abstract

P3: (real) dash repeat length

P4-P(n+4): (integer) list of n dash elements

nn LINE TYPE INITIAL OFFSET: has 1 parameter:

P1: (real) pattern offset

nn HATCH STYLE DEFINITION: has a variable parameter list:

P1: (index) hatch index
P2: (enumerated) style indicator: valid values are:

0 parallel
1 cross hatch

P3: (enumerated) hatch space units selector: valid values are:

0 VDC
1 proportion
2 fraction of display surface
3 abstract

P4: (vdc) first hatch direction vector x component
P5: (vdc) first hatch direction vector y component
P6: (vdc) second hatch direction vector x component
P7: (vdc) second hatch direction vector y component
P8: (real) duty cycle length
P9-P(9+n): (integers) list of gap widths
P(10+n)-P(10+2n): (integers) list of line widths
P(11+2n)-P(11+3n): (integers) list of n hatch elements
P(12+3n)-P(12+4n): (colors) list of line colors
P(12+4n)-P(12+5n): (enumerated) list ofd colour selection switches

nn LINE CAP: has 1 parameter:

P1: (index) line cap indicator: the following values are standardized:

1 unspecified
2 butt
3 round cap
4 projected square cap
5 triangle

P2: (enumerated) dash cap indicator: valid values are:

0 off
1 on

nn LINE JOIN: has 1 parameter:

P1: (index) line join indicator: the following values are standardized:

1 unspecified
2 miter
3 round
4 bevel

nn EDGE TYPE CONTINUATION: has 1 parameter:

P1: (index) continuation mode: the following values are standardized:

1 unspecified
2 continue
3 restart
4 adaptive continue

nn EDGE TYPE INITIAL OFFSET: has 1 parameter:

P1: (real) edge pattern offset

nn EDGE CAP: has 1 parameter:

P1: (index) edge cap indicator: the following values are standardized:

1 unspecified
2 butt
3 round cap
4 projected square cap
5 triangle

P2: (enumerated) dash cap indicator: valid values are:

0 off
1 on

nn EDGE JOIN: has 1 parameter:

P1: (index) edge join indicator: the following values are standardized:

1 unspecified
2 miter
3 round
4 bevel

nn MITER LIMIT: has 1 parameter:

P1: (real) miter limit

nn TEXT SCORE TYPE: has 1 parameter:

P1-Pn: list of score type, score indicator pairs (index,enumerated): the following values are standardized for the score type:

1   underscore
2   overscore
3   through score
4   kendot

valid values for the score indicators are:

0   off
1   on

nn   RESTRICTED TEXT TYPE: has 1 parameter:

P1: (index) restriction method: the following values are standardized:

1   basic
2   boxed
3   isotropic
4   justified
negative for private use

nn   GENERALIZED TEXT PATH MODE: has 1 parameter:

P1: (enumerated) text path mode: valid values are:

0   off
1   non-tangential
2   axis tangential

nn   SYMBOL LIBRARY INDEX: has 1 parameter:

P1: (index) symbol library index

nn   SYMBOL COLOUR: has 1 parameter:

P1: (colour) symbol colour

nn   SYMBOL HEIGHT: has 1 parameter:

P1: (vdc) symbol height

nn   SYMBOL ORIENTATION: has 4 parameters:

P1: (vdc) up vector x component
P2: (vdc) up vector y component
P3: (vdc) base vector x component
P4: (vdc) base vector y component

nn   GEOMETRIC PATTERN EXTENT: has 2 parameters:

P1: (point) first corner point
P2: (point) second corner point

nn   INTERPOLATED INTERIOR DEFINITION: has 5 parameters, the form of the colour parameters depends on the COLOUR SELECTION MODE:

P1: (point) first point
P2: (point) second point
P3: (colour) start colour
P4: (colour) end colour
P5: (index) style: the following values are standardized:

1	circular
2	parallel

nn PEL ARRAY COMPRESSION METHOD: has 1 parameter:

P1: (index) compression method: the following values are standardized:

1	T4
2	T6
3	LZW
4	bitmap
5	null background
6	null foreground

nn PEL ARRAY REFERENCE POINT: has 1 parameter:

P1: (point) reference point

Page 48

Add the following to the list of elements:

Class	Element Code	Element Name
0	6	BEGIN COMPOUND PATH
0	7	END COMPOUND PATH
0	8	BEGIN PROTECTED REGION
0	9	END PROTECTED REGION
0	10	BEGIN TILED PEL ARRAY
0	11	END TILED PEL ARRAY
0	12	BEGIN GEOMETRIC PATTERN
0	13	END GEOMETRIC PATTERN
1	19	COLOUR MODEL
1	20	FONT PROPERTIES
1	21	GLYPH MAPPING
1	22	COLOUR CALIBRATION
1	23	SYMBOL LIBRARY LIST
1	24	PICTURE MAPPING
3	13	PROTECTED REGION INDICATOR
3	14	DELETE PROTECTED REGION
4	22	HYPERBOLIC ARC
4	23	PARABOLIC ARC
4	24	NON-UNIFORM B-SPLINE

4	25	POLYBEZIER
4	26	PEL ARRAY
4	27	BITONAL PEL ARRAY
4	29	SYMBOL
5	36	LINE TYPE CONTINUATION
5	37	LINE AND EDGE TYPE DEFINITION
5	38	LINE TYPE INITIAL OFFSET
5	39	HATCH STYLE DEFINITION
5	40	LINE CAP
5	41	LINE JOIN
5	42	EDGE TYPE CONTINUATION
5	43	EDGE TYPE INTIAL OFFSET
5	44	EDGE CAP
5	45	EDGE JOIN
5	46	MITER LIMIT
5	47	TEXT SCORE TYPE
5	48	RESTRICTED TEXT TYPE
5	49	GENERALIZED TEXT PATH MODE
5	50	SYMBOL LIBRARY INDEX
5	51	SYMBOL COLOUR
5	52	SYMBOL HEIGHT
5	53	SYMBOL ORIENTATION
5	54	GEOMETRIC PATTERN EXTENT
5	55	INTERPOLATED INTERIOR DEFINITION
5	56	PEL ARRAY COMPRESSION METHOD
5	57	PEL ARRAY REFERENCE POINT

ANSI X3H3

Information Processing Systems --

Computer Graphics --

Metafile for the Storage and Transfer
of Picture Description Information

Part 4

Clear Text Encoding

Amendment 3

Draft Document 1.2

April 10, 1990

Page 11

Subclause 5.4.1: Add the following words to the deleted words list:

CURVE
MATRIX
NORMALIZED

Page 11

Subclause 5.4.3: Add the following words to the unabbreviated words list:

CAP
CONIC
BITONAL
BEZIER
FIT
GLYPH
JOIN
LIMIT
MAPPING
METHOD
MITRE
MODEL
OFFSET
PEL
POLYBEZIER
REGION
SCORE
SHIELD
SPLINE
SYMBOL
TILED

Page 12

Subclause 5.4.4: Add the following abbreviations:

ATTRIBUTES	ATTRS	
CALIBRATION	CALIB	
COMPOUND	COMPO	
COMPRESSION	CMPRSN	
CONTINUATION	CONT	
DEFINITION	DEF	
GENERALIZED	GEN	
GEOMETRIC	GEO	
HYPERBOLIC	HYPERB	?????
INITIAL	INIT	
INTERPOLATED	INTERP	
KERNING	KERN	
LIBRARY	LIB	
NON-UNIFORM B-SPLINE	NURB	?????
PARABOLIC	PARAB	?????
PROPERTIES	PROP	

PROTECTED	PROT
RATIONAL	RAT
TRANSFORMATION	TRAN
UNIFORM	UNIF

Page 13

Subclause 5.4.5: Add the following derived element names:

Metafile Name	Element Name	Notes
BEGIN COMPOUND PATH	BEGCOMPOPATH	
END COMPOUND PATH	ENDCOMPOPATH	
BEGIN PROTECTED REGION	BEGPROTREGION	
END PROTECTED REGION	ENDPROTREGION	
BEGIN TILED PEL ARRAY	BEGTILEDPELARRAY	
END TILED PEL ARRAY	ENDTILEDPELARRAY	
BEGIN GEOMETRIC PATTERN	BEGGEOPAT	
END GEOMETRIC PATTERN	ENDGEOPAT	
COLOUR MODEL	COLRMODEL	
FONT PROPERTIES	FONTPROP	
GLYPH MAPPING	GLYPHMAPPING	
COLOUR CALIBRATION	COLRCALIB	
SYMBOL LIBRARY LIST	SYMBOLLIBLIST	
PICTURE MAPPING	PICMAPPING	
PROTECTED REGION INDICATOR	PROTREGION	
DELETE PROTECTED REGION	DELPROTREGION	
HYPERBOLIC ARC	HYPERBARC	
PARABOLIC ARC	PARABARC	
NON-UNIFORM B-SPLINE	NURB	
POLYBEZIER	POLYBEZIER	
PEL ARRAY	PELARRAY	
BITONAL PEL ARRAY	BITONALPELARRAY	
SYMBOL	SYMBOL	
LINE TYPE CONTINUATION	LINETYPECONT	
LINE AND EDGE TYPE DEFINITION	LINEEDGETYPEDEF	
LINE TYPE INITIAL OFFSET	LINETYPEINITOFFSET	
HATCH STYLE DEFINITION	HATCHSTYLEDEF	
LINE CAP	LINECAP	
LINE JOIN	LINEJOIN	
EDGE TYPE CONTINUATION	EDGETYPECONT	
EDGE TYPE INITIAL OFFSET	EDGETYPEINITOFFSET	
EDGE CAP	EDGECAP	
EDGE JOIN	EDGEJOIN	
MITRE LIMIT	MITRELIMIT	
TEXT SCORE TYPE	TEXTSCORETYPE	
RESTRICTED TEXT TYPE	RESTRTEXTTYPE	
GENERALIZED TEXT PATH MODE	GENTEXTPATHMODE	
SYMBOL LIBRARY INDEX	SYMBOLLIBINDEX	
SYMBOL COLOUR	SYMBOLCOLR	
SYMBOL HEIGHT	SYMBOLHEIGHT	
SYMBOL ORIENTATION	SYMBOLORI	
GEOMETRIC PATTERN EXTENT	GEOPATEXT	
INTERPOLATED INTERIOR DEFINITION	INTERPINTDEF	

```
EL ARRAY COMPRESSION METHOD     PELARRAYCOMPRMETHOD
EL ARRAY REFERENCE POINT        PELARRAYREFPT
```

Page 15

Subclause 6.3: Add the following Metafile Descripter element encodings:

```
BEGIN COMPOUND PATH             ::= BEGCOMPOPATH
                                      <SOFTSEP>
                                      <TEXT|LINE>
                                    <TERM>

END COMPOUND PATH               ::= ENDCMPDPATH<TERM>

BEGIN PROTECTED REGION          ::= BEGPROTREGION
                                      <SOFTSEP>
                                      <I:REGIONINDEX>
                                      <SEP>
                                      <CLIP|SHIELD>
                                    <TERM>

END PROTECTED REGION            ::= ENDPROTREGION<TERM>

BEGIN TILED PEL ARRAY           ::= BEGTILEDPELARRAY
                                      <SOFTSEP>
                                      <I:FIRSTDIM>
                                      <SEP>
                                      <I:SECONDDIM>
                                      <SEP>
                                      <I:FIRSTOFFSET>
                                      <SEP>
                                      <I:SECONDOFFSET>
                                      <SEP>
                                      <I:FIRSTSIZE>
                                      <SEP>
                                      <I:SECONDSIZE>
                                    <TERM>

END TILED PEL ARRAY             ::= ENDTILEDPELARRAY<TERM>

BEGIN GEOMETRIC PATTERN         ::= BEGGEOPAT
                                      <SOFTSEP>
                                      <I:PATTERNINDEX>
                                    <TERM>

END GEOMETRIC PATTERN           ::= ENDGEOPAT<TERM>

COLOUR MODEL                    ::= COLRMODEL
                                      <SOFTSEP>
                                      <I:MODELINDEX>
                                    <TERM>

  PROPERTIES                    ::= FONTPROP
                                      <SOFTSEP>
```

```
                                   .
                                   .    * to be determined *
                                   .
                               <TERM>

GLYPH MAPPING                  ::= GLYPHMAPPING
                                   <SOFTSEP>
                                   <I:SETINDEX>
                                   <SEP>
                                   <BASISSET>
                                   <SEP>
                                   <I:OCTETSPERCODE>
                                   <SEP>
                                     .
                                     .   * to be determined *
                                     .
                               <TERM>

COLOUR CALIBRATION             ::= COLRCALBRTN
                                   <SOFTSEP>
                                   <R:XN>
                                   <SEP>
                                   <R:YN>
                                   <SEP>
                                   <R:ZN>
                                   <SEP>
                                   <R:XR>
                                   <SEP>
                                   <R:XG>
                                   <SEP>
                                   <R:XB>
                                   <SEP>
                                   <R:YR>
                                   <SEP>
                                   <R:YG>
                                   <SEP>
                                   <R:YB>
                                   <SEP>
                                   <R:ZR>
                                   <SEP>
                                   <R:ZG>
                                   <SEP>
                                   <R:ZB>
                               <TERM>

SYMBOL LIBRARY LIST            ::= SYMBOLLIST
                                   <OPTSEP>
                                   <S:SYMBOLNAME>
                                   <<SEP><S:SYMBOLNAME>>*
                               <TERM>

PICTURE MAPPING                ::= PICMAPPING
                                   <SOFTSEP>
```

```
                                        <R:X1>
                                        <SEP>
                                        <R:Y1>
                                        <SEP>
                                        <R:Z1>
                                        <SEP>
                                        <R:X2>
                                        <SEP>
                                        <R:Y2>
                                        <SEP>
                                        <R:Z2>
                                        <SEP>
                                        <R:X3>
                                        <SEP>
                                        <R:Y3>
                                        <SEP>
                                        <R:Z3>
                                      <TERM>

PROTECTED REGION INDICATOR      ::= PROTREGION
                                        <SOFTSEP>
                                        <I:REGIONINDEX>
                                        <SEP>
                                        <OFF|ON>
                                      <TERM>

L.   TE PROTECTED REGION        ::= DELPROTREGION
                                        <SOFTSET>
                                        <I:REGIONINDEX>
                                      <TERM>

HYPERBOLIC ARC              .   ::= HYPERBARC
                                        <SOFTSEP>
                                        <P:CENTREPOINT>
                                        <SEP>
                                        <P:TRANSVERSPOINT>
                                        <SEP>
                                        <P:CONJUGATEPOINT>
                                        <SEP>
                                        <VDC:STARTX>
                                        <SEP>
                                        <VDC:STARTY>
                                        <SEP>
                                        <VDC:ENDX>
                                        <SEP>
                                        <VDC:ENDY>
                                      <TERM>

PARABOLIC ARC                   ::= PARABARC
                                        <SOFTSEP>
                                        <P:TANGENTPOINT>
                                        <SEP>
                                        <P:STARTPOINT>
```

```
                                   <SEP>
                                   <P:ENDPOINT>
                                 <TERM>

NON-UNIFORM B-SPLINE             ::= NURB
                                   <SOFTSEP>
                                   <I:SPLINEORDER>
                                   <SEP>
                                     .
                                     .    * to be determined *
                                     .
                                   <SEP>
                                   <R:STARTVALUE>
                                   <SEP>
                                   <R:ENDVALUE>
                                 <TERM>


POLYBEZIER                       ::= POLYBEZIER
                                   <SOFTSEP>
                                     .
                                     .    * to be determined *
                                     .
                                 <TERM>

PEL ARRAY                        ::= PELARRAY
                                   <SOFTSEP>
                                   <I:PELID>
                                   <SEP>
                                   <0|90|180|270>
                                   <SEP>
                                   <90|270>
                                   <R:PELSPACING>
                                   <SEP>
                                   <R:LINESPACING>
                                   <SEP>
                                   <I:PELSPERLINE>
                                   <SEP>
                                   <I:LINES>
                                   <SEP>
                                   <I:COLORPRECISION>
                                   <SEP>
                                     .
                                     .    * to be determined *
                                     .
                                 <TERM>
BITONAL PEL ARRAY                ::= BITONALPELARRAY
                                   <SOFTSEP>
                                   <I:PELID>
                                   <SEP>
                                   <0|90|180|270>
                                   <SEP>
                                   <90|270>
```

```
                                   <R:PELSPACING>
                                   <SEP>
                                   <R:LINESPACING>
                                   <SEP>
                                   <I:PELSPERLINE>
                                   <SEP>
                                   <I:LINES>
                                   <SEP>
                                   <K:BACKGROUND>
                                   <SEP>
                                   <K:FOREGROUND>
                                   <SEP>
                                     .
                                     .    * to be determined *
                                     .
                                 <TERM>

SYMBOL                       ::= SYMBOL
                                   <SOFTSEP>
                                   <P:POINT>
                                   <SEP>
                                   <I:INDEX>
                                 <TERM>

LINE TYPE CONTINUATION       ::= LINETYPECONT
                                   <SOFTSEP>
                                   <I:CONTMODE>
                                 <TERM>

LINE AND EDGE TYPE DEFINITION ::= LINEEDGETYPEDEF
                                   <SOFTSEP>
                                   <I:LINETYPE>
                                   <SEP>
                                   <VDC|PROPORTION|FRACTION|ABSTRACT>
                                   <SEP>
                                   <R:REPEATLENGTH>
                                   <SEP>
                                     .
                                     .    * to be determined *
                                     .
                                 <TERM>

LINE TYPE INITIAL OFFSET     ::= LINETYPEINITOFFSET
                                   <SOFTSEP>
                                   <R:PATTERNOFFSET>
                                 <TERM>

HATCH STYLE DEFINITION       ::= HATCHSTYLEDEF
                                   <SOFTSEP>
                                   <I:HATCHINDEX>
                                   <SEP>
                                   <PARALLEL|CROSSHATCH>
                                   <SEP>
                                   <VDC|PROPORTION|FRACTION|ABSTRACT>
```

```
                                    <SEP>
                                    <VDC:FIRSTDIR>
                                    <SEP>
                                    <VDC:SECONDDIR>
                                    <SEP>
                                    <VDC:THIRDDIR>
                                    <SEP>
                                    <VDC:FOURTHDIR>
                                    <SEP>
                                    <R:DUTYCYCLE>
                                    <SEP>
                                      .
                                      .    * to be determined *
                                      .
                                  <TERM>

LINE CAP                      ::= LINECAP
                                    <SOFTSEP>
                                    <I:INDICATOR>
                                    <SEP>
                                    <OFF|ON>
                                  <TERM>

LINE JOIN                     ::= LINEJOIN
                                    <SOFTSEP>
                                    <I:INDICATOR>
                                  <TERM>

EDGE TYPE CONTINUATION        ::= EDGETYPECONT
                                    <SOFTSEP>
                                    <I:CONTMODE>
                                  <TERM>

EDGE TYPE INITIAL OFFSET      ::= EDGETYPEINITOFFSET
                                    <SOFTSEP>
                                    <R:PATTERNOFFSET>
                                  <TERM>
EDGE CAP                      ::= EDGECAP
                                    <SOFTSEP>
                                    <I:INDICATOR>
                                    <SEP>
                                    <OFF|ON>
                                  <TERM>

EDGE JOIN                     ::= EDGEJOIN
                                    <SOFTSEP>
                                    <MITRE|ROUND|BEVEL>
                                  <TERM>

MITRE LIMIT                   ::= MITRELIMIT
                                    <SOFTSEP>
                                    <R:MITRELIMIT>
                                  <TERM>
```

```
     SCORE TYPE                   ::= TEXTSCORETYPE
                                        <SOFTSEP>
                                          .
                                          .     * to be determined *
                                          .
                                      <TERM>

RESTRICTED TEXT TYPE              ::= RESTRTEXTTYPE
                                        <SOFTSEP>
                                        <I:TEXTFIT>
                                      <TERM>

GENERALIZED TEXT PATH MODE        ::= GENTEXTPATHMODE
                                        <SOFTSEP>
                                        <OFF|NONAXIS|AXIS>
                                      <TERM>

SYMBOL LIBRARY INDEX              ::= SYMBOLINDEX
                                        <SOFTSEP>
                                        <I:INDEX>
                                      <TERM>

SYMBOL COLOUR                     ::= SYMBOLCOLR
                                        <SOFTSEP>
                                        <K:SYMBOLCOLOUR>
                                      <TERM>

SYMBOL HEIGHT                     ::= SYMBOLHEIGHT
                                        <SOFTSEP>
                                        <VDC:HEIGHT> (non-negative)
                                      <TERM>

SYMBOL ORIENTATION                ::= SYMBOLORI
                                        <SOFTSEP>
                                        <DELTAPAIR> (up vector)
                                        <SEP>
                                        <DELTAPAIR> (base vector)
                                      <TERM>

GEOMETRIC PATTERN EXTENT          ::= GEOPATEXT
                                        <SOFTSEP>
                                        <P:FIRSTPOINT>
                                        <SEP>
                                        <P:SECONDPOINT>
                                      <TERM>

INTERPOLATED INTERIOR DEFINITION::= INTERPINTDEF
                                        <SOFTSEP>
                                        <P:FIRSTREFPOINT>
                                        <SEP>
                                        <P:SECONDREFPOINT>
                                        <SEP>
                                        <K:FIRSTCOLOUR>
                                        <SEP>
```

```
                                        <K:SECONDCOLOUR>
                                        <SEP>
                                        <I:STYLE>
                                      <TERM>

PEL ARRAY COMPRESSION METHOD      ::= PELARRAYCMPRSNMETHOD
                                        <SOFTSEP>
                                        <I:METHOD>
                                      <TERM>

PEL ARRAY REFERENCE POINT         ::= PELARRAYREFPT
                                        <SOFTSEP>
                                        <P:REFPOINT>
                                      <TERM>
```

FINAL REPORT

CALS CY90 SOW TASKS 4.2.1 AND 4.2.2

PRODUCE TOOL TO DETERMINE CONFORMANCE
OF A FILE TO CGM FIPS 128 AND TO MIL-D-28003

PURPOSE

Produce a tool to determine conformance of a file to CGM FIPS (Task 4.2.1) and to MIL-D-28003 (Task 4.2.2)

BACKGROUND

The final deliverable for these tasks was completed by the subcontractor and has been delivered to NIST/CSL. NIST/CSL has provided comments on previous versions of the software (both the Alpha and Beta versions), and has influenced the final deliverable. As stipulated in the contract, this software tool provides only one copy which may be used internally at NIST/CSL. It is not available for distribution by NIST/CSL at this time. The tool is undergoing final acceptance by NIST/CSL now.

The software tool, called CTS/METACALS by the vendor, is a stand-alone program that interprets data files conforming to the CGM standard and the CALS AP (MIL-D-28003). It produces a report detailing how well a given CGM conforms to the standard and the CALS AP (MIL-D-28003).

Deliverables to CALS DoD in previous years have led to this development. First, a plan for such development was created. Then a comprehensive list of tasks and responsibilities was prepared. Based on one item in that list, the development of a test method, a Test Requirements Document was created. The software tool was programmed to meet the specifications of that document.

DISCUSSION

This software tool runs on an IBM PC 286 or compatible equivalent. With the tool in place, NIST/CSL can begin testing CALS instances of CGMs against the standard and the AP. NIST/CSL requires examples of metafiles from CALS for inhouse testing purposes.

A mutually agreeable distribution license may then be worked out between the vendor and NIST/CSL (if NIST/CSL should establish a licensing agreement with the vendor).

CALS USE/IMPACT

This tool should provide CALS the necessary mechanism to ensure reliability of CGM interchange of illustration data in the CALS environment.

RECOMMENDATIONS

NIST/CSL recommends that the tool be thoroughly tested and debugged before any decisions are made concerning its use in the CALS environment. DOD, through CALS office approval, should provide some examples of metafiles for NIST/CSL inhouse testing. Further, the next step in this progression calls for a tool to be developed to test generators of CGM metafiles. This tool would be able to test generators to ensure that they always produce metafiles which conform to both the standard and the CALS AP.

FINAL REPORT

CALS CY90 SOW TASK 4.2.3

EXPLORE SOURCES OF CGM GENERATOR/INTERPRETER CONFORMANCE TEST CAPABILITIES

PURPOSE

Explore potential sources of generator/interpreter conformance test capabilities. (Task 4.2.3)

BACKGROUND

It was known from the beginning of the CALS work back in 1986 that the Computer Graphics Metafile (CGM) standard offered no conformance statements concerning either writers (generators) or readers (interpreters) of metafiles. An international workshop on CGM Certification held in the United Kingdom in March of 1987 concluded that "a CGM Testing Architecture must include testing for CGM generators and interpreters."

Work performed by the NIST Graphics Software Group for the CALS Program since that time has concentrated on developing the Application Profile for CGM in CALS, namely MIL-D-28003. Over the last three years NIST has developed a testing methodology to test metafiles both to the level of the CGM standard and to the level of MIL-D-28003. A test tool has been developed, and testing is due to begin shortly. The next logical phase of this work involves developing conformance tests for generators.

A necessary first step in this development is to explore all potential sources of generator/interpreter conformance test capabilities.

DISCUSSION

Work done for this task in FY90 has been to formulate a detailed outline of a **Business Plan for Marketing the Conformance Testing of Graphics Standards** (Appendix 1), including that of CGM. In addition a **Certification Survey Form** (Appendix 2) has been sent to appropriate vendors and users in the graphics marketplace, around twenty-five all together.

If the CALS Policy Office has other contacts who should be filling out this Survey Form, please forward their names and addresses to NIST.

CALS USE/IMPACT

This initial business plan outline and survey are the necessary first steps in providing the information that CALS needs to make informed decisions concerning the questions of a CGM Certification Laboratory for CGM generators writing metafiles conforming to both the CGM standard and the MIL-D-28003.

RECOMMENDATION

In CY91 NIST will report results from the survey and interpret those results in light of CALS testing requirements.

APPENDIX 1

A DETAILED OUTLINE OF A
BUSINESS PLAN FOR MARKETING THE
CONFORMANCE TESTING OF GRAPHICS STANDARDS

Part I -- Background

A. What are the Graphics Standards

1. Data Interchange Formats
    a. Computer Graphics Metafile (CGM)
2. Application Programmer Interface Standards
    a. Graphical Kernel System (GKS)
    b. Programmers' Hierarchical Interactive Graphics System (PHIGS and PHIGS PLUS)
    c. Computer Graphics Interface (CGI) Language Bindings
3. Device Interface Protocols
    a. Computer Graphics Interface (CGI) Data Stream Encodings
4. The Register of Graphical Items
    a. ISO TR 9973
    b. The Role of NIST as Registration Authority

B. Brief Survey of Testing Tools Available

1. CGM Testing Tools
2. PHIGS Testing Tools
3. GKS Testing Tools
4. CGI Testing Tools

C. Guidelines for Conformance Testing

1. ISO DIS 10641, Conformance Testing of Implementations of Graphics Standards
2. FIPSPUB titled, "GOSIP (Government Open Systems Interconnection Profile) Conformance and Interoperation Testing and Registration"
3. OSI Testing Strategy
    a. National Voluntary Laboratory Accreditation Program
    b. NIST Certificate of Validation Testing
    c. OSINET
    d. Technical Oversight Committee
    e. Testing and Registration Service

Part II -- Marketing Considerations

A. Conformance Testing Services

1. What are the Services
2. Who needs them
3. What are the specific benefits
4. Market size estimates
   a. NCGA Integrate Survey Results
   b. NIST Survey Results

B. Role of a Testing Laboratory

1. Accreditation
2. Services
3. Operation
4. Oversight
   a. Responsibility
   b. Testing Control Boards
5. Financials
   a. Expenses
   b. Income
6. Publicity

C. Conclusions

1. Initially concentrate on CGM
2. Focus on CALS, but also support general CGM testing for the non-DOD sector

Part III --Marketing Strategy for a NIST/CSL Testing Laboratory

A. Testing Services

1. CGM Testing
   a. CGM testing to ISO 8632
   b. CGM testing to the CALS CGM AP (MIL-D-28003)
   c. Testing CGM Generators for Conformance to the CALS AP
   d. Testing CGM Interpreters for Conformance to the CALS AP
2. PHIGS Testing to ISO 9592-1/ISO 9593-1 (Fortran)
   a. Data Structure Tests
   b. Error Tests
   c. Operator Tests
   d. Archive File Tests (Parts 2 & 3)
   e. Provisions for Additional Language Bindings (C and Ada)
   f. Provisions for PHIGS PLUS Testing (Part 4)
3. GKS Testing to ISO 7942/ISO 8805-1 (Fortran)
   a. Data Structure Tests
   b. Error Tests
   c. Operator Tests
   d. Provisions for Additional Language Bindings (C and Ada)

4. CGI Testing to ISO 9636/ISO 9738-4 (C)
   a. Data Structure Tests
   b. Error Tests
   c. Operator Tests
   d. Provisions for Additional Language Bindings (Fortran)
   e. Provisions for Data Stream Encoding Testing (ISO 9837)

B. Education and Training Services

1. Short Courses
2. Handbook for Contracting Officers
3. Workshops and User Groups

Part III --Marketing Strategy for a NIST/CSL Testing Laboratory (Continued)

C. Operational Issues

1. Equipment
2. Staffing
3. Publication of Results
   a. By Testing Lab
   b. By the Client
4. Legal Considerations
   a. Confidentiality
   b. Client licensing of test tools
5. Certification
   a. Initial Testing
   b. Retesting
6. Test Suite Maintenance
7. Dispute Resolution
   a. Control Board
   b. Certification Body

D. Publicity/Advertising Methods

1. Press Releases and Publications
2. Trade Shows and Demonstrations
3. Intragovernmental Contacts

E. Financials

1. Income
2. Expenses
   a. Labor
   b. Equipment (hardware and software)
   c. Administrative Costs (Mailings, Consumables, etc.)
3. Cash Flow Model

F. Other Marketing Strategies and Tactics

1. Membership/Sponsorship

APPENDIX 2

CGM TESTING AND CERTIFICATION SURVEY FORM

NOTE: On the following survey form, trade names for specific products are used. Inclusion of a specific product in no implies a recommendation or endorsement by NIST. Similarly, the omission of a particular product does not imply that its capabilities are less than those of the included products.

CGM Testing and Certification Survey Form
(Make a new copy of this entire survey for each product)

1. <u>Background Data</u>.

(1A) Company: ____________________

(1B) Address: ____________________

(1C) ____________________

(1D) ____________________

(1E) Technical POC: ____________________

(1F) Telephone: ____________________

(1G) FAX: ____________________

(circle one)

(1H) Are you a supplier of a product that incorporates a CGM capability or an end-user of such a product? Supplier / End-User

<u>Product Incorporating a CGM Capability</u>

(1I) Product Name: ____________________

(1J) Product Type (circle one):

(1) Word Processing (2) Presentation Graphics (3) File Converter Utility
(4) Graphics Library (5) CAD application (6) Electronic Publishing
(7) Other (describe): ____________________

(1K) CGM Capability (circle one or both): (1) Imports CGMs (2) Exports CGMs

(1L) Platform/OS Supported:

(1L1)	MS/PC-DOS	Yes	No
(1L2)	Unix Workstations	Yes	No
(1L3)	VAX/VMS	Yes	No
(1L4)	IBM mainframe	Yes	No

Other: ____________________ (1M1) (platform) ____________________ (1M2) (operating system)

Other: ____________________ (1M3) (platform) ____________________ (1M4) (operating system)

Other: ____________________ (1M5) (platform) ____________________ (1M6) (operating system)

2. <u>General Information</u>. (Note: If you are an end user, rather than a supplier, interpret the marketing aspects of the following questions as questions about your use of this product.)
On a scale of 1 to 5 (circle one for each question):

	not important				very important
(2A) How important is the CALS market to your sales strategy for this product?	1	2	3	4	5
(2B) How important is the CGM capability in this product to its success?	1	2	3	4	5
(2C) How important would it be to your product's success to be awarded a certificate of conformance?	1	2	3	4	5

If your product EXPORTS CGMs:

(2D) How important is it to your product's success that you export correct CGMs?	1	2	3	4	5
(2E) How important is it to your product's success that other products (yours or someone else's) can correctly import and render for viewing and/or hardcopy the CGMs exported by your product?	1	2	3	4	5

Which other products (rate each one)?

(2E1) Aldus Pagemaker	1	2	3	4	5
(2E2) Xerox Ventura Publisher	1	2	3	4	5
(2E3) Interleaf	1	2	3	4	5
(2E4) Other: ______________________________	1	2	3	4	5

If your product IMPORTS CGMs:

(2G) How important is it to your product's success that you import and correctly render all CGMs conforming to the CALS Application Profile (MIL-D-28003)?	1	2	3	4	5
(2H) How important is it to your product's success that it correctly imports and render for viewing and/or hardcopy CGMs exported by other products?	1	2	3	4	5

Which other products (rate each one)?

<u>PC Products</u>

(2I1) Harvard Graphics	1	2	3	4	5
(2I2) Lotus Freelance	1	2	3	4	5
(2I3) Genigraphics	1	2	3	4	5
(2I4) Pansophic	1	2	3	4	5
(2I5) Autographix	1	2	3	4	5
(2I6) Other: ______________________________	1	2	3	4	5
(2I7) Other: ______________________________	1	2	3	4	5
(2I8) Other: ______________________________	1	2	3	4	5

Apple Macintosh Products

(2K1)	Aldus Persuasion	1	2	3	4	5
(2K2)	Microsoft Powerpoint	1	2	3	4	5
(2K3)	GSC GraphPorter	1	2	3	4	5
(2K4)	Other: ______________________________	1	2	3	4	5
(2K5)	Other: ______________________________	1	2	3	4	5
(2K6)	Other: ______________________________	1	2	3	4	5

Other Platforms

(2L1)	Advanced Technology Center GKS	1	2	3	4	5
(2L2)	Computer Associates DISPLA	1	2	3	4	5
(2L3)	Digital Equipment Corporation GKS/PHIGS	1	2	3	4	5
(2L4)	Hewlett-Packard Starbase	1	2	3	4	5
(2L5)	Precision Visuals DI-3000	1	2	3	4	5
(2L6)	Sun Microsystems SunGKS/SunGKS	1	2	3	4	5
(2L7)	Other: ______________________________	1	2	3	4	5
(2L8)	Other: ______________________________	1	2	3	4	5
(2L9)	Other: ______________________________	1	2	3	4	5

In your market for your specific product:

(2M) How many competitors' products are there?

1-2	3-6	7-10	more than 10

(2N) Approx. what percentage of your competitors' products also offer a CGM capability?

no more than 25%	no more than 50%	no more than 75%	more than 75%

List some of your competitors' products that incorporate a CGM capability:

(2P1) Company: ____________________ Product: ____________________

(2P2) Company: ____________________ Product: ____________________

(2P3) Company: ____________________ Product: ____________________

(2P4) Company: ____________________ Product: ____________________

(2P5) Company: ____________________ Product: ____________________

(2P6) Company: ____________________ Product: ____________________

(2P7) Company: ____________________ Product: ____________________

3. Conformance Testing and Certification

Discussion. Two kinds of "correctness testing" for CGMs can be imagined. Syntactic correctness refers to an absence of violations of the standard (ISO 8632; ANSI/X3.122; FIPS 128) and the CALS Application Profile (MIL-D-28003). Semantic correctness refers to an absence of errors in representing (by using the proper sequence of CGM elements) the picture intended by the system that generated the CGM. Given these definitions, a testing and certification process might work something like this:

The testing lab (either a private lab or a government-run lab) receives a CGM file or product for testing. As a result of the testing process, a test report listing all errors found is produced and sent to the submitter. If no errors are found, the testing laboratory or some related certification body issues a certificate for the specific version of the product. If errors are found, the submitter may provide a revised version of the product and request a retest. This process may iterate until no errors are detected by the testing lab.

In the questions that appear in this section, you will be asked to indicate the **MOST THAT YOU WOULD BE WILLING TO PAY** and the **LONGEST TIME THAT YOU WOULD BE WILLING TO WAIT** for various testing services proposed in the question. When answering the question, please keep in mind the following assumptions: (1) retesting will be performed at a reduced rate proportional to the amount of effort required for retesting and (2) if travel to the submitter's site is required, costs will be increased by about $250/day of testing plus round-trip airfare.

So, with the above definitions and discussion in mind, please indicate how much you would be willing to pay for various testing services.

(3A1) Single instances of CGMs for CALS syntactic correctness?	no more than $100	no more than $175	no more than $250	more than $250

What would be an acceptable turnaround time between:

(3A2) receipt of CGM and sending of test report?	no more than 1 day	no more than 3 days	no more than 5 days	more than 5 days
(3A3) sending of test report and receipt of certificate?	no more than 1 wk	no more than 2 wks	no more than 4 wks	more than 4 weeks

(3B1) Single instances of CGMs for both CALS syntactic and semantic correctness?	no more than $200	no more than $350	no more than $500	more than $500

What would be an acceptable turnaround time between:

(3B2) receipt of CGM and sending of test report?	no more than 1 dy	no more than 3 dys	no more than 5 dys	more than 5 days
(3B3) sending of test report and receipt of certificate?	no more than 1 wk	no more than 2 wks	no more than 4 wks	more than 4 weeks

(3C1) Products exporting CGMs for CALS syntactic correctness?	no more than $2K	no more than $5K	no more than $10K	more than $10K

What would be an acceptable turnaround time between:

(3C2) installation of product at lab and sending of test report?	no more than 1 wk	no more than 2 wks	no more than 4 wks	more than 4 weeks
(3C3) sending of test report and receipt of certificate?	no more than 6 wks	no more than 9 wks	no more than 12 wks	more than 12 weeks
(3D1) Products exporting CGMs for both CALS syntactic and semantic correctness?	no more than $5K	no more than $8.5K	no more than $10K	more than $10K

What would be an acceptable turnaround time between:

(3D2) installation of product at lab and sending of test report?	no more than 2wks	no more than 4 wks	no more than 6 wks	more than 6 weeks
(3D3) sending of test report and receipt of certificate?	no more than 6 wks	no more than 9 wks	no more than 12 wks	more than 12 weeks
(3E1) Products for their ability to import all CALS CGMs and render them correctly?	no more than $10K	no more than $15K	no more than $20K	more than $20K

What would be an acceptable turnaround time between:

(3E2) installation of product at lab and sending of test report?	no more than 4wks	no more than 6 wks	no more than 8 wks	more than 8 weeks
(3E3) sending of test report and receipt of certificate?	no more than 6 wks	no more than 9 wks	no more than 12 wks	more than 12 weeks

What methods would you use to transmit CGMs to the testing lab?

(3F1)	PC 5¼" floppy diskette	Yes	No
(3F2)	PC 3½" diskette	Yes	No
(3F3)	Mac 3½" diskette	Yes	No
(3F4)	electronic mail	Yes	No
(3F5)	VAX/VMS cartridge tape	Yes	No
(3F6)	UNIX tar tape	Yes	No
(3F7)	ANSI magnetic (reel) tape	Yes	No
(3F8)	Other: ______________________		

Should a testing certificate be a requirement for delivering CGM instances to the government or to subcontractors in a CALS-compliant procurement?

(3G1) YES NO
If no, why not? ______________________________
(3G2) ______________________________

Should a testing certificate be a requirement for delivering products capable of importing or exporting CALS CGMs in a CALS-compliant procurement?

For CALS CGM exporters (circle one)?

(3H1) YES NO
If no, why not? ______________________________
(3H2) ______________________________

For CALS CGM importers (circle one)?

(3I1) YES NO
If no, why not? ______________________________
(3I2) ______________________________

Would you be willing to pay for CGM Testing and Certification of the CGM encodings other than the binary encoding used by CALS?

(3J1) Character coding? YES NO

(3J2) If yes, how much?

less than for binary encoding	same as for binary encoding	more than for binary encoding

(3K1) Clear text? YES NO

(3K2) If yes, how much?

less than for binary encoding	same as for binary encoding	more than for binary encoding

NIST-114A (REV. 3-90)	U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY	1. PUBLICATION OR REPORT NUMBER NISTIR 4579
	BIBLIOGRAPHIC DATA SHEET	2. PERFORMING ORGANIZATION REPORT NUMBER
		3. PUBLICATION DATE MAY 1991

4. TITLE AND SUBTITLE

NIST Support for the Computer-aided Acquisition and Logistic Support (CALS) Program in the Area of Graphics Standards, Calendar Year 1990.

5. AUTHOR(S)

Daniel R. Benigni, Editor

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)	7. CONTRACT/GRANT NUMBER
U.S. DEPARTMENT OF COMMERCE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY GAITHERSBURG, MD 20899	8. TYPE OF REPORT AND PERIOD COVERED NISTIR 10/89 thru 12/90

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

Office of the Secretary of Defense
Production and Logistics/Systems/CALS
Room 3B322, Pentagon
Washington, DC 20301-8000

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

Computer-aided Acquistion and Logistic Support (CALS) is a program of the Office of the Secretary of Defense. Its objective is to establish an integrated set of standards and specifications for the creation, management, and exchange of logistics data and product development data--including graphical data--by computer. Since FY86, NIST has been funded to recommend the standards to satisfy CALS requirements for system integration and digital data transfer, and to accelerate standards implementation. This report comprises the continuing work of the NIST Graphics Software Group in support of the CALS Program for CY90. The format of this report combines the separate task deliverables assigned to the NIST Graphics Software Group for CY90, which were as follows: (1) update CGM Application Profile (i.e., update MIL-D-28003); (2) inject CALS requirements into the standards committees' work on Amendments to the CGM standard; (3) produce a software tool to determine conformance of a metafile to the CGM standard and to MIL-D-28003; (4) and explore potential sources of generator and interpreter conformance test capabilities.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

application profile; CALS; CGM; CGM Amendment 1; CGM Amendment 3; conformance testing; DoD; graphics; illustration data; MIL-D-28003; MIL-D-28003A

13. AVAILABILITY		14. NUMBER OF PRINTED PAGES
XX	UNLIMITED	316
	FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).	15. PRICE
	ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.	A14
X	ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.	

ELECTRONIC FORM

HB