NISTIR 4775

Incorporating CALS Requirements into the CGM Standard and the CALS Application Profile – MIL-D-28003

Daniel R. Benigni

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

U.S. DEPARTMENT OF COMMERCE
Rockwell A. Schnabel, Acting Secretary

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

NIST

NISTIR 4775

Incorporating CALS Requirements into the CGM Standard and the CALS Application Profile – MIL-D-28003

Daniel R. Benigni

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

February 1992

U.S. DEPARTMENT OF COMMERCE
Rockwell A. Schnabel, Acting Secretary

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

INCORPORATING CALS REQUIREMENTS
INTO THE CGM STANDARD AND THE
CALS APPLICATION PROFILE--MIL-D-28003

TABLE OF CONTENTS

EXECUTIVE SUMMARY 1

BACKGROUND 2

ACKNOWLEDGMENT. 3

APPENDIX 1: MIL-D-28003A. 5

APPENDIX 2: CGM AMENDMENT 3 TEXT. 57

APPENDIX 3: CGM AMENDMENT 4 ANSI PROJECT PROPOSAL 259

APPENDIX 4: CGM AMENDMENT 4 TEXT 267

EXECUTIVE SUMMARY

Early in the Computer-aided Acquisition and Logistic Support (CALS) initiative, NIST proposed and CALS adopted the Computer Graphics Metafile (CGM) standard for the exchange of illustration data. NIST realized at the time that CALS requirements mandated changes to CGM because of recognized shortcomings. The military specification known as MIL-D-28003 was written by NIST to address some of those shortcomings.

Additional CALS requirements studies by NIST pointed to areas where the CGM standard itself needed to be extended with additional functionality. CALS-sponsored work was started by NIST in both the American and International standards organizations to add this functionality to CGM.

This standards work has led to Amendment 3 of CGM (Appendix 1), which is now almost complete. It will be folded into a completely revised national/international CGM standard to be published in mid-1992. The functionality for Amendment 3 is still undergoing change as it is folded into the revised CGM standard. The revision of MIL-D-28003, known as MIL-D-28003A (Appendix 2), incorporates the functionality of CGM Amendment 3. The CGM standard with this amendment, together with MIL-D-28003A, can now be used in CALS to exchange engineering drawing data as well as illustration data. CALS should consider the use of MIL-D-28003A as an option to IGES (Initial Graphics Exchange Specification) for the interchange of engineering drawings in certain situations.

In addition, work has also commenced on another CGM amendment concerning rules for profiles and conformance. This activity was not anticipated a year ago, but it will have significant impact on the evolution and successful use of MIL-D-28003A. The ISO/IEC CGM committee recognized that among the principal practical barriers to automated graphical interchange are:

- A lack of sensible rules for profiles of CGM (MIL-D-28003A is a profile); and

- A lack of a foundation in the standard itself for testing conformance of CGM generators and interpreters.

Accordingly, a new project (Appendix 3 contains the project proposal) was initiated (known as CGM Amendment 4) to define new normative parts of CGM which define rules for profiles and conformance requirements for CGM software components. The US provided the initial text for CGM Amendment 4 (Appendix 4). The work heavily reflects previous CALS profile work, and in turn goes further. This work will affect future revisions of 28003.

BACKGROUND

After six years of deliberation, circulation, balloting, and refinement the Computer Graphics Metafile (CGM) became an American National Standards Institute (ANSI) standard in 1986, and an ISO/IEC standard in 1987. (ISO/IEC stands for International Organization for Standardization/International Electrotechnical Commission.) One consequence of the consensus process of drafting and refining the CGM is that the standard tended toward a "least common denominator" graphical metafile for the various constituents. It is, to a large degree, the area of overlap that all participants in its formulation agreed should be in a graphical metafile. As a result, it is functionally lean. The advantage is that CGM has been easier to implement and has come into wide use.

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 elements of the sort needed in compound document development and 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 extension process commenced, sorting through the proposals to enrich CGM functionality in the direction of more advanced metafile applications. As a result, three amendments (formerly called "addenda") to CGM were initiated:

1. Amendment 1 - additional output capabilities and symbol libraries;

2. Amendment 2 - support of 3-dimensional (3D) primitives and viewing in metafiles;

3. Amendment 3 - support of advanced 2-dimensional (2D) drawing capabilities for technical illustration, graphics art quality picture definition, and graphics in technical publishing.

All three were initiated as ISO/IEC projects. The first two were endorsed in 1986 by vote of ISO/IEC JTC1/SC24 (Joint Technical Committee 1/Sub-Committee 24), and the third by the formal requirements and NWI (New Work Item) process of ISO/IEC JTC1/SC24, during the period 1987-1989. In 1989 the project finally passed the SC24 NWI ballot, principally as a consequence of CALS participation. SC24 finally approved NWI status for CGM Amendment 3 in 1989 and ISO/IEC JTC1 concurred in 1990.

ANSI procedures now automate adoption of these projects as ANSI standards, so there is no separate ANSI technical effort (or rather, the ANSI efforts are brought to bear as positions in the ISO/IEC project). Furthermore, early in 1991, ANSI gave final approval to withdraw the U.S. version of CGM, ANS X3.122-1986 (FIPS PUB 128 is being modified as a result and is currently out for public review), and replace it with the technically identical ISO/IEC version. As a US standard, it is now designated ANSI/ISO 8632.

Amendment 3 is the most important for CALS. Parts of Amendment 1 are useful as well. Amendment 2 is not relevant to CALS. Other standards, such as IGES (the Initial Graphics Exchange Specification) and STEP (Standard for Exchange of Product Model Data), offer 3D capability that does not need to be repeated in CGM. The US position has been that the work should be terminated because there is little use for Amendment 2, it confuses implementors, and it drains scarce resources from Amendment 3 work.

Due in part to the CALS project, in 1987 the scope of Amendment 1 was defined to include functionality important to CALS constituents (e.g., Global Segments). Amendment 3 was started within ANSI primarily due to CALS efforts. In fact, progress on Amendment 3 is principally due to CALS contributions.

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/IEC was sought. The result was a study group, formed to determine the need for the Amendment, generate a requirements statement, and produce a New Work Item proposal. The NWI ballot passed in July 1989 and the circulation and review of technical Working Drafts commenced.

Amendment 1 was published by ISO/IEC in November 1991, and has now been adopted by ANSI.

Amendment 3 completed three review cycles and two major processing milestones in 1990. Early in 1991 Amendment 3 began its final step - the Draft AMendment (DAM) ballot - on schedule for completion in summer 1991 and publication in the fall. This publication schedule would have been maintained, but during the year the ISO/IEC CGM committee decided that Amendment 3 would not be published as a separate ISO/IEC standard. Rather, Amendment 1, Amendment 3, and the large number of resolved Defect Reports would be folded into the base standard and the single document defining CGM versions 1, 2 and 3 would be published. The single remaining task to date is for the document editor to produce: 2nd draft ISO/IEC review text; and the final text (following ISO/IEC review).

In addition to these CGM amendments, functionality important to CALS was submitted to the Graphical Registration process during the period 1986-1988. This functionality, in the form of items to be registered through the Graphical Registration process, was promoted and coordinated with Amendment 3 work throughout 1988 and 1989. In 1990 the Amendment 3 work, which overlaps the registration work significantly, caught up with and moved ahead of the registration work in the ISO/IEC processing (due to very slow and cumbersome procedures for registration).

In 1991, as in 1990, the nature of the work was not so much generating and injecting new CALS requirements into the CGM amendments, but expediting the progress of the amendments and ensuring that they in fact meet the approved requirements. Both of these activities have been critical. The adherence of Amendment 3 to the approved scope and requirements had to be monitored and enforced, or the schedule would slip. Keeping Amendment 3 on schedule in 1991 was critical because of its tight coordination with the review and publication of MIL-D-28003A (which includes Amendment 3) by November 1991.

In 1991 the ISO/IEC committee (SC24/WG3) initiated work to define rules and guidelines for application profiles for CGM (such as MIL-D-28003), and to add metafile generator and interpreter conformance requirements to the CGM standard.

ACKNOWLEDGMENT

The author wishes to acknowledge the efforts of Lofton R. Henderson, Henderson Software, and Lynne Rosenthal, NIST. They have represented NIST at CGM-related national and international standards meetings during the past year, and have ensured that CALS requirements have been incorporated into the CGM Amendment work at those meetings.

APPENDIX 1

MIL-D-28003A

NON-MEASUREMENT SENSITIVE

MIL-D-28003A
15 November 1991

SUPERSEDING
MIL-D-28003
20 December 1988

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 the 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 Classification. 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) specified herein consists of three parts: the metafile, the generator (which writes the metafile), and the interpreter (which reads the metafile). A metafile or an interpreter shall be classified as one of the following types:

Type 0--monochrome;

Type 1--grayscale;

Type 2--full color.

Metafiles conforming to any of these types are called "conforming basic metafiles." Interpreters conforming to any of these types are called "conforming basic interpreters." Generators that produce conforming basic metafiles are called "conforming basic generators."

2. APPLICABLE DOCUMENTS

2.1 Government documents.

2.1.1 **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.

FEDERAL INFORMATION PROCESSING STANDARD

FIPS PUB 128 - Computer Graphics Metafile (CGM)

Note: FIPS PUB 128 adopts ANSI/ISO 8632:1992 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 Standardization Documents Order Desk, Building 4D, 700 Robbins Avenue, Philadelphia, PA 19111-5094. Others must request copies of FIPS from the National Technical Information Service, 5285 Port Royal Road, Springfield, VA 22161.)

MILITARY STANDARD

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

(Copies of the referenced military standard are available from the Standardization Documents Order Desk, Building 4D, 700 Robbins Avenue, Philadelphia, PA 19111-5094.)

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 8632-1:1992 Computer Graphics Metafile (CGM), Part 1, Functional Specification

ISO 8632-3:1992 Computer Graphics Metafile (CGM), Part 3, Binary Encoding

NOTE: This is the revision of ISO 8632:1987 defining version 1, version 2, and version 3 metafiles.

(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 the 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. The conformance requirements specified herein pertain to the conformance of CGM metafiles, generators, and interpreters. CGM elements and parameters allowed in a conforming metafile are specified herein. The CGM metafile may include all elements and parameters specified in FIPS PUB 128 that are not disallowed by this specification, but to meet the requirements of a conforming basic metafile the values of CGM elements and parameters shall be restricted to the values in the "Basic Set" as specified herein. Illustration data which meets the requirements of this specification shall be in the form of one or more conforming basic metafiles.

3.1.1 Conforming basic metafile. A conforming basic metafile shall:

a. Conform to one of the versions 1, 2, or 3 as defined in Clause 7 of ANSI/ISO 8632:1992;
b. Conform to one of the three types (Type 0, 1, or 2) classified herein;
c. Contain no scalar values of parameter data outside the ranges specified herein;
d. Be encoded only with the CGM Binary Encoding, as defined in FIPS PUB 128 (ANSI/ISO 8632:1992, part 3); and
e. Conform to any additional metafile requirements specified herein.

3.1.2 Conforming basic generator. A conforming basic generator shall:

a. Be a "conforming MIL-D-28003A generator" according to the rules of Clause 7 of ANSI/ISO 8632:1992;
b. Produce only conforming basic metafiles (or can be reliably commanded to function in that mode);
c. Accurately implement all of the functional elements and parameters of FIPS PUB 128 [including the guidelines of FIPS PUB 128 (ANSI/ISO 8632:1992, annex D)];
d. Generate metafiles which accurately and correctly define the intended picture; and
e. Conform to any additional generator requirements specified herein.

3.1.3 Conforming basic interpreter. A conforming basic interpreter shall:

a. Be a "conforming MIL-D-28003A interpreter" according to the rules of Clause 7 of ANSI/ISO 8632:1992;

b. At a minimum, correctly interpret any conforming basic metafile of the same type (Type 0, 1, or 2);
[Note: A Type 1 interpreter shall correctly interpret Type 0 and 1 basic conforming metafiles, and a Type 2 interpreter shall correctly interpret Type 0, 1, and 2 basic conforming metafiles.]
c. Accurately implement all of the functional elements and parameters of FIPS PUB 128 [including the guidelines of FIPS PUB 128 (ANSI/ISO 8632:1992, annex D.2 and D.5), and the recommendations for the treatment of circular and elliptical graphical primitive elements of FIPS PUB 128 (in ANSI/ISO 8632:1992, annex D.4.5] except as modified herein;
d. Conform to the additional interpreter requirements as specified herein;
e. Generate pictures which accurately and correctly represent the metafile being interpreted;
f. Parse and skip the elements and parameter values specified herein as ignorable (when interpreting a non-conforming metafile);
g. Render all text at "stroke" precision, regardless of the value of the metafile TEXT PRECISION element; and
h. Render color exactly, according to the category of the interpreter: monochrome, grayscale, or color.

Note: In cases where a contract allows mapping of metafile color or grayscale (e.g., many metafile colors to fewer or different interpreter colors, or color to monochrome) the following principles shall be applied:

(1) If the metafile color selection mode is "direct":
  (a) the value of the metafile BACKGROUND COLOUR shall map to one of the device background colors;
  (b) any color value of any other metafile element that is exactly equal to the value of the metafile BACKGROUND COLOUR shall also map to the device background color; and
  (c) all other color values in the metafile shall map to another device color that must be distinct from the device background color, and that must be closest to the specified metafile color according to some reasonable metric applied to color space.

(2) If the metafile color selection mode is "indexed":
  (a) only the BACKGROUND COLOUR and COLOUR TABLE elements contain RGB values to be mapped; thus, the metafile "effective background color" shall be specified 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;
  (b) the effective background color shall map to one of the device background colors;
  (c) any COLOUR TABLE values that exactly match the effective background color shall also map to this value; and
  (d) all other RGB values shall map to another device color that must be distinct from the device background color, and that must be closest to the specified metafile color according to some reasonable metric applied to color space.

3.1.4 **Physical file structure.** If the delivery medium is magnetic tape, then the metafile data shall be blocked into records of 800 bytes. For other media, the format shall be as specified in MIL-STD-1840A or by procurement specification.

3.1.5 **Defects in FIPS PUB 128.** A number of editorial defects (or errors) have been found in ANSI/ISO 8632, the standard adopted by FIPS PUB 128.

3.2 **Specific requirements.** Specific requirements for conforming basic metafiles, generators, and interpreters are specified in the following subsections. Conforming basic metafiles shall meet the constraints on the elements of FIPS PUB 128 as specified herein.

3.2.1 **Metafile constraints.** The Basic Set of values shall be limited by the constraints on Basic values and additional values as specified below. Where an element or parameter of FIPS PUB 128 is not mentioned herein, the Basic Set shall include all values of that element or parameter as specified in FIPS PUB 128.

3.2.1.1 **Delimiter elements.** Delimiter elements shall meet the constraints specified in table I. There are no other parameter range constraints imposed upon the delimiter elements, other than the length limits and Name Precision limits specified in 3.2.1.2.

TABLE I. Delimiter element constraints

Element	Basic Values
Version 1 Elements	
No additional constraints	
Version 2 Elements	
BEGIN SEGMENT	(Note 1)
END SEGMENT	
Version 3 Elements	
BEGIN COMPOUND LINE	(Note 2)
END COMPOUND LINE	
BEGIN PROTECTION REGION	
END PROTECTION REGION	
BEGIN TILE ARRAY	
END TILE ARRAY	(Note 3)

Note 1: Constraints on segments include constraints on the number of simultaneously defined segments and the nesting of segments specified herein.

Note 2: Constraints on the number and size of contributing elements shall be as for CLOSED FIGURE (see TABLE V).

Note 3: Constraints on tiled raster arrays include constraints on tile size and number of tiles as specified herein.

3.2.1.2 **Metafile descriptor elements.** Metafile descriptor elements shall meet the constraints specified in table II.

TABLE II. Metafile descriptor element constraints

Element	Basic Values
Version 1 Elements	
METAFILE VERSION	1,2,3 (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 4,7) (1,4/1) (Note 5,7)
CHARACTER CODING ANNOUNCER	0 (Basic 7-bit) 1 (Basic 8-bit)
MAXIMUM COLOUR INDEX	(Note 6)
Version 2 Elements	
NAME PRECISION	8,16
MAXIMUM VDC EXTENT	no constraints
SEGMENT PRIORITY EXTENT	no constraints
Version 3 Elements	
COLOUR MODEL	element not allowed
COLOUR CALIBRATION	element not allowed
FONT PROPERTIES	element not allowed
GLYPH MAPPING	element not allowed
SYMBOL LIBRARY LIST	element not allowed

Note 1: Any of the version values: "1", "2", and "3" 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"; and

c) shall have appended to this latter string either:
".0" if the metafile is monochrome;
".1" if the metafile is grayscale; or
".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 3.2.4.4.

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: For color metafiles, Basic values shall be limited to 0-255; for grayscale metafiles, Basic values shall be limited to 0-15; for monochrome metafiles, Basic values shall be limited to 0-1. MAXIMUM COLOUR INDEX shall apply to all color indexes defined or otherwise referenced, whether they are referenced implicitly or explicitly. 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.

Note 7: 4/2 and 4/1 are "column/row" notation for positions in code tables. These designate the character codes with decimal numeric values "66" and "65" respectively. The parameters are the 1-character strings "B" and "A" respectively.

3.2.1.3 Picture descriptor elements. Picture descriptor elements shall meet the constraints specified in table III.

TABLE III. Picture descriptor element constraints

Element	Basic Values
Version 1 Elements	
COLOUR SELECTION MODE	(Note 1)
SCALING MODE	(Note 2)
LINE WIDTH SPECIFICATION MODE	(Note 3)
MARKER SIZE SPECIFICATION MODE	(Note 3)
EDGE WIDTH SPECIFICATION MODE	(Note 3)
Version 2 Elements	
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
Version 3 Elements	
INTERIOR STYLE SPECIFICATION MODE	(Note 3)
LINE & EDGE TYPE DEFINITION	(Note 4)
HATCH STYLE DEFINITION	(Note 5)
GEOMETRIC PATTERN DEFINITION	element not allowed

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

Note 2: It is a rule of ANSI/ISO 8632:1992 that the scale-factor parameter of SCALING MODE shall always be 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 real numbers have been selected: its precision shall be (0,9,23).

Note 3: All of the values "scaled," "absolute," "fractional," and "mm" of ANSI/ISO 8632:1992 shall be Basic values.

Note 4: The number of entries in the dash gap list parameter shall not exceed 8. At most 16 user line types shall be specified simultaneously.

Note 5: The number of entries in the gaps array list parameter shall not exceed 8. At most 16 user hatch styles shall be specified simultaneously.

3.2.1.4 **Control elements.** Control elements shall meet the constraints specified in table IV.

TABLE IV. Control element constraints

Element	Basic Values
Version 1 Elements	
VDC INTEGER PRECISION	16, 32
VDC REAL PRECISION	(1,16,16) (fixed) (0,9,23) (floating point)
TRANSPARENCY	1 (on)
Version 2 Elements	
LINE CLIP MODE	1 (shape) (Note 1)
MARKER CLIP NODE	1 (shape) (Note 1)
EDGE CLIP MODE	1 (shape) (Note 1)
NEW REGION	no constraints
SAVE PRIMITIVE CONTEXT	element not allowed
RESTORE PRIMITIVE CONTEXT	element not allowed
Version 3 Elements	
MITRE LIMIT	no constraints
PROTECTION REGION INDICATOR	element not allowed
GENERALIZED TEXT PATH MODE	element not allowed
TRANSPARENT CELL COLOUR	no constraints

Note 1: Because the single allowed value is not the default value, this element shall appear in every conforming metafile, either in the picture body or in a Metafile Defaults Replacement, if the corresponding primitive is present in the metafile.

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. The parameter lists of graphical primitive elements shall meet the constraints specified in Table V. [Note: In the table "npts" refers to the number of points in a point list.]

TABLE V. Graphical primitive constraints

Element	Basic Values
Version 1 Elements	
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)
Version 2 Elements	
Closed Figure	(Note 3)
CONNECTING EDGE	no constraints
CIRCULAR ARC CENTER REVERSED	no constraints
Version 3 Elements	
HYPERBOLIC ARC	no constraints
PARABOLIC ARC	no constraints
POLYBEZIER	1..256 Bezier segments
BITONAL TILE	(Note 4)
NON UNIFORM B-SPLINE	order, 2..7; number of control points, < or = 1024
NON UNIFORM RATIONAL B-SPLINE	order, 2..7; number of control points, < or = 1024

Note 1: In addition, any sub-polygon shall be well defined and have a minimum of 3 points.

Note 2: The string parameters of graphical text shall not contain any control characters (7 or 8 bit codes in the ranges 1..31 and 128..159) except as allowed by and necessary to implement the character set switching modes which can be selected by Basic values of CHARACTER CODING ANNOUNCER. The C0 character NUL (code value 0) shall be permitted, and shall have no effect. A string with one or more NUL characters present is exactly equivalent to the same string with those characters removed.

Note 3: Constraints on this element include those specified elsewhere herein on the individual components of this element (e.g., polyline vertex constraints). The number of individual graphical primitive elements comprising the Closed Figure shall not exceed 32.

Note 4: The Basic values for the bitonal tile compression type parameter shall be limited to 0-6. Constraints on the size and number of tiles which are specified elsewhere herein shall also apply.

3.2.1.6 **Attribute elements.** Attribute elements shall meet the constraints specified in table VI.

TABLE VI. Attribute element constraints

Element	Basic Values
Version 1 Elements	
LINE BUNDLE INDEX	1-5
LINE TYPE	1-5, 6-15 (Note 1)
LINE WIDTH	positive (Note 2)
MARKER BUNDLE INDEX	1-5
MARKER TYPE	1-5
MARKER SIZE	positive
TEXT BUNDLE INDEX	1-2
TEXT FONT INDEX	1-32 (Note 3,4)
CHARACTER HEIGHT	positive
CHARACTER SET INDEX	1-2 (Note 3,5)
ALTERNATE CHARACTER SET INDEX	1-2 (Note 3,5)
FILL BUNDLE INDEX	1-5
HATCH INDEX	1-6
EDGE BUNDLE INDEX	1-5
EDGE TYPE	1-5
EDGE WIDTH	positive (Note 2)
PATTERN TABLE	Index, 1-8 nx, 1-16 ny, 1-16
COLOUR TABLE	(Note 6)
Version 2 Elements	
PICK IDENTIFIER	(Note 7)
Version 3 Elements	
LINE CAP	1-5 and unspecified/match
LINE JOIN	1-4
LINE TYPE CONTINUATION	1-4
LINE TYPE INITIAL OFFSET	[0.0,1.0] (Note 8)
TEXT SCORE TYPE	element not allowed
RESTRICTED TEXT TYPE	2-6 (Note 9)

TABLE VI. Attribute element constraints - Continued.

Element	Basic Values
Version 3 Elements - Continued	
INTERPOLATED INTERIOR	element not allowed
EDGE CAP	1-5 and 1-3
EDGE JOIN	1-4
EDGE TYPE CONTINUATION	1-4
EDGE TYPE INITIAL OFFSET	[0.0,1.0] (Note 8)
SYMBOL LIBRARY INDEX	element not allowed
SYMBOL COLOUR	element not allowed
SYMBOL SIZE	element not allowed
SYMBOL ORIENTATION	element not allowed

Note 1: The line types specified in 3.2.2.1 shall be included in the Basic Set, and comprise the registered index values 6-15. These values have been registered with the ISO Registration Authority for Graphical Items and are in accordance with the ISO Register of Graphical Items.

Note 2: The width shall not exceed 10% of the drawing size, which for this purpose is defined as the shortest side of the UDC Extent. Wider lines shall be rendered as filled areas.

Note 3: The character set selected shall be representable in the font selected.

Note 4: Every referenced font index shall correspond to a defined entry in the FONT LIST.

Note 5: Every referenced character set index shall correspond to a defined entry in the CHARACTER SET LIST.

Note 6: For color metafiles, the start index shall be 0-255; for grayscale metafiles, the start index shall be 0-15; for monochrome metafiles the start index shall be 0-1.

Note 7: 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.

Note 8: The notation means 0.0 to 1.0 inclusive.

Note 9: The default value of this element is 1, which is not included in the Basic Set. This element shall appear in every conforming basic metafile which uses RESTRICTED TEXT.

3.2.1.7 **Segment elements.** Segment elements shall meet the constraints specified in table VII.

TABLE VII. Segment element constraints

Element	Basic Values
Version 2 Elements	
COPY SEGMENT	(Note 1)
INHERITANCE FILTER	no constraints
CLIP INHERITANCE	element not allowed
SEGMENT TRANSFORMATION	no constraints
SEGMENT HIGHLIGHTING	element not allowed
SEGMENT DISPLAY PRIORITY	no constraints
SEGMENT PICK PRIORITY	no constraints (Note 2)

Note 1: Segment copy references shall not be nested more than 4 deep. That is, the depth of the hierarchy implied by allowing the COPY SEGMENT function to appear within the definition of another segment shall not exceed 4. Depth 1 shall correspond to flat structure, i.e., no nesting or hierarchy.

Note 2: 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 specified in the Metafile Descriptor, all global segment definitions shall follow all other Metafile Descriptor elements. When global segments are specified in the Picture Descriptor (Version 3 metafiles only), all global segment definitions shall follow all other Picture Descriptor elements.

3.2.1.7 **ESCAPE element.** To ensure portability and predictability of results, conforming metafiles shall contain only those ESCAPE elements that are specified in 3.2.4.5 herein.

3.2.1.8 **External elements.** No constraints. However, the "action required" flag of the MESSAGE element shall not be used in such a way that the picture definition is altered.

3.2.2 **Additional attribute values.**

3.2.2.1 **Line types.** Additional line types permitted under this specification are specified in table VIII.

TABLE VIII. Additional line types

LINE TYPE	ISO Register 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 **Element defaults.** The defaults of all elements shall be as specified in FIPS PUB 128 (ANSI/ISO 8632, clause 6 of Part 1). Conforming basic metafiles shall be permitted to contain one or more METAFILE DEFAULTS REPLACEMENT elements to redefine any of these values with values from the Basic Set.

3.2.4 **Semantic ambiguities.** FIPS PUB 128 leaves the semantics of a number of graphical details unspecified or "implementation dependent." Requirements in the following sections shall apply for conforming basic generators and interpreters.

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

3.2.4.2 **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 the CLIP INDICATOR 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.

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

3.2.4.4 **Font specifications.** The fonts in Table IX are public domain fonts, available as part of NBS SP 424. All of these fonts shall be basic capabilities of a conforming basic metafile. Any of these fonts may appear in the FONT LIST element in a conforming basic metafile. 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. Font name shall be designated as in Table IX. The string "HERSHEY:" shall be an acceptable substitute for "HERSHEY/".

TABLE IX. 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	
xx.	HERSHEY/SYMBOL_SET_1	(Note 2)
xx.	HERSHEY/SYMBOL_SET_2	(Note 2)
xx.	HERSHEY/SYMBOL_MATH	(Note 2)

Note 1: Code tables defining the association of numeric character code value with character (glyph) will be included in a future revision of this specification.

Note 2: The set of required glyphs and the codes to invoke them will be included in a future revision of this specification.

TABLE X contains additional font names included in the Basic Set of fonts. The fonts in TABLE IX and TABLE X together shall comprise the Basic Set of the FONT LIST element. Any of these font names may appear in the FONT LIST element in a basic conforming metafile, and basic conforming metafiles shall not reference any font not listed in TABLE IX or TABLE X. A basic conforming interpreter may substitute fonts metrically identical to these named fonts when rendering a basic conforming metafile.

Some of the font names in TABLE X are trademarked. Some of the named fonts are proprietary and copyrighted, and therefore require permission of the owners to use them. However, this specification in no way requires the license of named fonts from their trademark or copyright owners. Metric equivalents of the named fonts are widely available. Substitution by interpreters of fonts which are "metrically equivalent", as specified in 4.3.1, constitutes compliance.

TABLE X. More basic font names

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

Note 1: Times and Helvetica are registered trademarks of Allied Corporation, the owner of the copyright on the fonts of those names.

Note 2: The "SYMBOL" font in TABLE X contains greek characters in the familiar alphabetic positions, and various mathematical and publishing symbols in the upper code positions (159-255). The set of required glyphs and the codes to invoke them will be included in a future revision of this specification.

The case (upper/lower) of the font names of the above font tables in FONT LIST elements shall not be significant to conforming basic generators and conforming basic interpreters.

3.2.4.5 **Escape elements.** Conforming basic interpreters shall support the following ESCAPE elements:

None.

3.2.5 **Implementation requirements for conforming basic generators and interpreters.** The requirements in this section augment those of FIPS PUB 128 (ANSI/ISO 8632, Part 1, annex D.5, and Part 3, clause 8). These requirements specify additional element constraints and Basic values for certain maxima and minima that shall apply to conforming basic metafiles.

Name: METAFILE DEFAULTS REPLACEMENT

Description: The METAFILE DEFAULTS REPLACEMENT element shall not be partitioned. Elements within METAFILE DEFAULTS REPLACEMENT 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.

Name: COLOUR TABLE

Description: 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 specified and used, it shall not be respecified. [Note: These restrictions insure that interpreting systems without dynamic color update capabilities shall be able to render the intended picture accurately.]

Note 1: For indexed color selection, either background color and all color indexes in the metafile shall have their representations specified or none shall. Color indexes shall be specified by the COLOUR TABLE element. Background color shall be specified either by the BACKGROUND COLOUR element or by the color index 0 (BACKGROUND COLOUR is synonymous with color index 0--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 color is automatically "used" upon the occurrence of BEGIN PICTURE BODY.

Note 2: For direct color selection, either the background color and the color of each displayed primitive shall be explicitly specified, or none shall. In other words, either all colors shall be defaulted or none shall.

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. A PATTERN TABLE element defining the representation of a given pattern index shall be present if that pattern index is used within the picture. It shall appear before explicit reference to that index by any PATTERN INDEX element; or 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 with interior style "pattern." Once a given pattern representation is specified and used, it shall not be respecified. [**Note:** These restrictions insure that interpreting systems without dynamic pattern update capabilities shall be able to render the intended picture accurately.]

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 element (one 1024x1024 image); 256 for each PATTERN TABLE element (one 16x16 pattern); and 2048 for the complete pattern table itself (eight 16x16 patterns); for Type 2 (color) metafiles, 256 for each COLOUR TABLE element (entries 0-255), and 256 for the complete color table itself; for Type 1 (grayscale) metafiles, 16 for each COLOUR TABLE element (entries 0-15) and 16 for the complete color table itself; for Type 0 (monochrome) metafiles, 2 for each COLOUR TABLE element (entries 0-1) and 2 for the complete color table itself; CELL ARRAY and PATTERN TABLE have color array parameters and COLOUR TABLE has a color list parameter.

Name: Tile Array

Description: Tile Array consists of tiles of compressed color specifiers. Each tile shall be no larger than 1048576 compressed color specifiers (one 1024x1024 image); there shall be no more than 256 tiles.

Name: Maximum Point Array Length

Description: The Basic value for the maximum 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 maximum 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 herein. To insure predictable results, conforming interpreters and generators shall use the default values from Table XI.

TABLE XI. Default bundle tables

Bundle Type	Bundle Index 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 shall be 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.** Deliverables under this specification shall meet all requirements of section 3. The inspection set forth herein shall become a part of the contractor's overall inspection system or quality program. The absence of any inspection requirements specified herein 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.** Conforming basic metafiles shall be analyzed for conformance to the Basic Set of values for CGM elements and parameters of FIPS PUB 128 and as specified in section 3 herein. To confirm that the specifications of FIPS 128 and section 3, herein have been met, CGM Validation Test Software has been developed and a CGM Validation Test Service has been established by the Computer Systems Laboratory at the National Institute of Standards and Technology (NIST). The contractor shall be responsible for obtaining validation of the CGM deliverable from this NIST Validation Test Service. Conforming basic generators shall be examined as necessary to ascertain that they generate only conforming basic metafiles, and those metafiles correctly represent the picture within the constraints imposed herein. Conforming basic interpreters shall be examined as necessary to ascertain that they meet all of the minimum requirements specified herein, and that they produce the correct picture.

4.3.1 **Font rendering.** Any rendering of a font as specified herein is conforming if the rendering is "metrically identical" to the font metrics of the requested font. That is, 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. Such rendering 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 specified herein shall be met 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.

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 the interpreter shall not "crash," and if possible that element shall 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 as specified herein represents illustration data in the form of a conforming basic metafile, i.e., it contains, in device-independent, system-independent, and implementation-independent form, the picture description data represented by the functions invoked through an application program interface. This specification defines the allowable elements and parameters which may be used to compose the picture. In addition, certain constraints on CGM generators and interpreters are specified herein to remove implementation dependencies, thereby serving to ensure predictable interchange of conforming basic metafiles between clients. Thus, this specification may also be used in a contract to define the technical requirements to be met when it is desired to purchase conforming basic generators and/or interpreters.

6.1.1 **CGM Application Profile (AP).** Most standards tend to be very completely specified, and ideally all implementations of standards would implement all possible elements, options and permissible values. Then there would be no problems in exchanging either implementations or files. However, not all standards are completely specified, nor are all possible elements, options and permissible values implemented by all implementors. FIPS PUB 128, the CGM standard, is not completely specified. This was intentional on the part of the CGM standards committee to allow implementation on a wider range of existing systems and make the CGM standard more adaptable to the various needs and philosophies of a diverse clientele. This introduces difficulties in trying to unambiguously describe an intended picture using the CGM standard. Nor have all implementors of the CGM standard chosen to include all possible elements, options and permissible values available and allowed in the standard. For their own reasons they have chosen particular subsets of the CGM standard to implement. This means that a picture written to a metafile using a particular generator may not be able to be fully rendered using a particular interpreter. Unpredictable results can and do occur. In addition, since the behavior of generators and interpreters themselves are not part of the CGM standard, a further unpredictability of results can occur.

This is the reason that certain groups of users have gotten together to rigorously define and adhere to the same subset of the CGM standard. This ensures predictable results and inter-working between machines, sites, and applications. Such subsets are known as application profiles.

This specification is a CGM Application Profile for a particular group of users, namely DoD and contractors dealing with DoD. It is based on FIPS PUB 128, and defines the allowable elements, parameters and options which may be used to compose a picture in a metafile. In addition to more completely specifying semantic gaps in the CGM standard, specifying the operations and required capabilities of generators and interpreters, specifying the particular subset of CGM elements and parameters and their Basic Set of values, this specification also:

- specifies implementation requirements;
- specifies maxima and minima values of certain CGM elements and parameters; and
- specifies some additional element values which have completed the process of Graphical Registration.

6.1.2 **Structure of the CGM standard and this specification.** Metafiles in the CGM standard are defined as a series of layers of detail. The highest level of structure is the metafile itself. Each metafile may contain one or more pictures, which are completely independent of each other. Each of the items stored in a metafile is stored as an element. Each element may have associated with it a list of data called parameters. Elements may be grouped into Segments, which may be referenced and reused multiple times. There are nine classes of elements, and each is defined in turn in the CGM standard: Delimiter elements, Metafile Descriptor elements, Control elements, Graphical Primitive elements, Attribute elements, Escape elements, External elements, and Segment elements. Specific requirements in section 3 herein are organized similarly. Additionally, a new class of elements is specified in section 3 herein, namely Segment elements, and are inserted after Attribute elements.

6.1.3 **Basic and permissible values.** "Permissible values" are the range of values for CGM elements and parameters as specified in FIPS PUB 128. "Basic values" are the range of permissible values that are mandatory for conformance to this specification, and are specified herein. In some cases Basic values are augmented herein by additional values. Thus, both the Basic values and additional values specified herein constitute the "Basic Set."

6.1.4 **FIPS PUB 128.** FIPS PUB 128 adopts the international Computer Graphics Metafile (CGM) standard, ANSI/ISO 8632, as the Federal Information Processing Standard (FIPS) for use by Federal agencies. The FIPS only provides for the use of ANSI/ISO 8632 within the Federal government, explaining what it is and how it is to be applied and implemented. ANSI/ISO 8632, part 1, provides the functional specification for CGM elements and parameters, and their permissible values, while parts 2, 3 and 4 specify encodings. In particular, ANSI/ISO 8632, part 3, specifies the Binary encoding. All references to FIPS PUB 128 herein apply to ANSI/ISO 8632, and where necessary will also cite in parentheses the particular reference location in ANSI/ISO 8632.

6.1.5 **Metafile Descriptor Elements.** It is unclear in FIPS PUB 128 whether there should be a mandatory ordering of Metafile Descriptor elements (the grammar implies some). ANSI/ISO 8632 recommends such an ordering; METAFILE VERSION, METAFILE ELEMENT LIST, and METAFILE DESCRIPTION should be the first three elements, in that order.

6.1.6 **Additional attribute values.**

6.1.6.1 **Line types.** The line types specified in table VIII 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 VIII, 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.

6.1.6.2 **Fonts and Character Sets.** 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. 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 of the requested Hershey font. For example, the style and metrics of a Hershey version of the character "LOWER CASE A ACCENT GRAVE" should have an obvious relationship to those of "LOWER CASE A."

This problem does not arise in the other font families specified herein.

It is recognized that the Hershey fonts may not be of adequate quality for modern publication requirements.

6.2 Ordering data. The contract or purchase order should specify the following:

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

b. Whether the metafile is Type 0 (monochrome), Type 1 (grayscale), or Type 2 (full color). (See 1.2 and 3.2.1.2)

c. Physical file structure (see 3.1.4)

6.3 Definitions.

6.3.1 Acronyms and abbreviations used herein. Acronyms and abbreviations used herein 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 Standard.

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

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

g. **ISO** - International Organization for Standardization.

h. **PUB** - Publication.

i. **RGB** - The color model Red-Green-Blue.

j. **SP** - Special Publication.

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

6.3.2 **Application Profile.** A set of specifications (beyond that in the published standard) appropriate to a particular environment. The goal of an AP is to eliminate implementation dependencies and provide for the effective and unambiguous use of a standard.

6.3.3 **Basic set.** The set of Basic values and additional values for CGM elements and parameters as specified herein.

6.3.4 **Basic values.** Basic values are the subset of permissible values that are mandatory for conformance to this specification, and are specified herein.

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

6.3.6 **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 specified herein.

6.3.7 **Conforming basic interpreter.** A metafile interpreter that correctly interprets and renders any conforming basic metafile of the same conformance classification (Type 0, 1, or 2) as specified herein, and additionally conforms to any additional requirements as specified herein.

6.3.8 **Conforming basic metafile.** A metafile that complies with this specification and that conforms to one of the three categories classified herein. Type 0 metafiles shall be specified as monochrome metafiles; Type 1 metafiles shall be specified as grayscale metafiles; and Type 2 metafiles shall be specified as color metafiles.

6.3.9 **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.

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

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

6.3.12 **Permissible values.** The range of values for CGM elements and parameters as specified in FIPS PUB 128.

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

Note: Refer to FIPS PUB 128 (i.e., ANSI/ISO 8632, 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

10. SCOPE.

10.1 **Scope.** This appendix is provided for informative purposes, to assist in evaluating revision MIL-D-28003A. It provides a compendium of the differences between the original specification and Revision A of the specification. This appendix is not a mandatory part of this specification. The material contained herein is intended for guidance only.

20. **APPLICABLE DOCUMENTS.** This section is not applicable to this appendix.

30. **BACKGROUND.** MIL-D-28003 was completed and published as a part of the CALS Phase I program for digital delivery of technical materials in 1988. As explained in section 6, MIL-D-28003 is an "Application Profile" of the CGM standard, ANSI/ISO 8632:1987. It is the way in which the CGM standard is harnessed and applied for use by the CALS application community.

At the time that the 28000-series specifications were being devised, there was a similar initiative underway in the manufacturing sector--MAP/TOP. Work was well underway for a CGM application profile for this sector. There was close collaborative effort between this community and the CALS community to improve the quality of the specifications and to devise a single profile to serve the needs of both communities. The CGM profile of MAP/TOP V3.0 and MIL-D-28003 were very close in content and structure. The goal for the CALS community for this initial profile was a basic usable profile which did not deviate significantly from the profile of the MAP/TOP community.

During this period and subsequently the CALS program conducted requirements studies to more carefully define the needed facilities in a graphics format intended for technical illustrations. The results of these studies indicated a number of extensions and modifications specifically needed by the CALS constituency. Work commenced to produce a revision of MIL-D-28003 that more closely reflected the needs of the CALS community.

Concurrently work progressed within the ANSI and ISO graphics standards communities on two amendments to CGM, the so-called Amendment 1 and Amendment 3, which respectively defined Version 1 and Version 3 metafiles. There was also work within the standards communities to give public standing to certain extensions to the CGM standard (and other graphics standards as well) by the process of "graphical registration". Finally, there was work in the ISO CGM committee, SC24/WG3 to resolve outstanding "defect reports" on CGM.

These revisions to the base CGM standard comprise additions that in many cases have been driven by the needs of constituencies such as technical publishing, engineering drawing, and graphic arts. Amendment 1 completed processing in late 1990, and Amendment 3 completed processing in fall of 1991. The two amendments plus the defects resolutions were folded into ISO 8632:1987 to produce ISO 8632:1992 (and the identical ANSI/ISO designations).

40. **OVERVIEW OF CHANGES IN MIL-D-28003A.** MIL-D-28003A contains additions which include standardized functionality from CGM Amendment 1 (Version 2 metafiles) and the nearly complete CGM Amendment 3 (Version 3 metafiles).

Besides additions MIL-D-28003A also contains some deletions and some changes. The experience of the CALS community and CGM standard users in general has provided information necessary to fine tune some of the original specifications of MIL-D-28003.

50. **CRITERIA FOR CHANGES TO THE PROFILE.** A number of criteria were considered in evaluating potential changes to MIL-D-28003. In many cases these criteria conflict and tradeoffs had to be made.

50.1 **Universal Printability.** It should be possible to build implementations based on commonly available technology that can print conforming metafiles as specified herein. Esoteric or unusual resources should not be required or allowed (e.g., unusual private font collections should be proscribed, as they tend to isolate islands of automation that have access to them).

50.2 **Uniformity of Results.** The specification should be sufficiently unambiguous that uniform results can be obtained based on this published specification in combination with the CGM standard itself. Each feature should be evaluated, and implementation leeway in rendering the feature should only be allowed if it is a specifically desirable feature.

50.3 **High Expressive Power.** This specification should enable conforming metafiles to efficiently represent the graphical features commonly used in technical publishing. There should be good support for graphical text, advanced curves and conics, reusable symbols, etc.

50.4 **Implementability.** The additions to MIL-D-28003 should accomplish the most important currently-unsatisfied requirements of technical publishing without unduly increasing the implementation burden. Some increase is inevitable, but lower priority additions requiring extra work disproportionate to their utility should be deferred until a future revision (see Appendix B).

50.5 **Close to Base Standard.** As much as possible the specification should consist of limitations of the base CGM standard and specifications of implementation dependencies. Redefinitions of the base standard must absolutely be avoided, and unusual extensions should only be included where strongly justified. It is necessary that any conforming basic metafile be a legal CGM, and desirable that it be largely understood by high quality complete interpreters, even if they are not specifically written to comply with this specification.

50.6 **Avoid Excessive Subsetting and Levels.** Experience with standards and military specifications has shown that each conformance subset or level tends to become a separate dialect and attract a separate clique. Not only does universal printability suffer, but there is potential for widespread misunderstanding among consumers of conforming products as to what the product actually is.

60. **SPECIFIC CHANGES IN MIL-D-28003A.** The following summarizes the most important modifications, additions, and deletions in deriving MIL-D-28003A from MIL-D-28003. As an aid to evaluating these changes, the rationale for each change is discussed briefly.

60.1 **Draft and Publication Conformance Levels.** Some experts view the distinction between Draft and Publication qualities of interpreters as not useful. Digital deliverables will require publication-quality interpreters for either softcopy or hardcopy operational use. A draft quality interpreter will certainly be a useful tool in many circumstances, but is not identified as a key operational requirement for the digital delivery of documents. Therefore the concept of these two distinct conformance levels has been removed from MIL-D-28003A.

60.2 **METAFILE VERSION.** The Basic values of this element have changed from "1" (the only value in ANSI/ISO 8632:1987) in MIL-D-28003 to "1," " 2," or "3" in MIL-D-28003A (all of the values allowed in ANSI/ISO 8632:1992). In addition to reflecting the set of elements that may be present, the Version also may reflect changes in grammar. For example Version 2 metafiles allow the COLOUR TABLE to be present in the picture descriptor as well as the picture body, and in fact this is a preferable location.

60.3 Color.

60.3.1 Conformance Levels. A very high proportion of technical illustrations will be monochrome, or at best grayscale. MIL-D-28003 could be read as requiring that any conforming interpreter be able to print wide-spectrum full color. Features have been added to MIL-D-28003A to allow a conforming basic metafile to be classified and self-identified as Type 0 (monochrome), Type 1 (grayscale), or Type 2 (full color), and conforming basic interpreters can conform to one of those three categories.

60.3.2 **MAXIMUM COLOR INDEX.** The requirements have been aligned with the three conformance categories, and an ambiguity in the ANSI/ISO 8632 standard regarding the applicability to "implicitly defined" color indexes has been clarified.

60.3.3 Default Color Table. The default color table to be assumed by conforming basic interpreters in the absence of explicit color information has been deleted. There is no longer a specified default. The specification was contrary to the spirit of the specification that "all color table entries shall be defined or none shall." The "none" subclause was specifically intended to allow interpreter leeway in selection of colors where the precise colors either did not matter to the generator, or where the appropriate set have depended upon such factors as hardcopy vs. softcopy presentation of documents. In general the latter freedom will be useful for monochrome applications which make up the majority of technical illustrations.

60.3.4 All-or-none for Direct Color. MIL-D-28003 only applied the all-or-none definition requirement to the indexed color selection method. MIL-D-28003A extends the same principle to direct color selection.

60.3.5 Include BACKGROUND COLOUR in All-or-none. The indexed color "all-or-none" definition requirement of MIL-D-28003 overlooked that the BACKGROUND COLOUR element behaves the same as the definition of an index by COLOUR TABLE. It has now been included appropriately.

60.3.6 **Color Table Size.** The various requirements regarding starting index, table size, etc, have been modified to reflect the three conformance types of Type 0 (monochrome), Type 1 (grayscale), and Type 2 (full color).

60.4 **METAFILE DESCRIPTION.** The string will now reflect that the file conforms to MIL-D-28003A. The requirements for inclusion of identifying information for the source of the metafile are increased. A new suffix to the string identifies to which of the three conformance categories the file belongs.

60.5 **Additional Attribute and Primitive Restrictions.** A number of situations that are indeterminate in the ANSI/ISO 8632 base standard are specified:

1. The value 0 is prohibited for LINE WIDTH, EDGE WIDTH, and MARKER SIZE.

2. POLYLINEs with less than 2 points encoded, POLYGONs (and sub-polygons of POLYGON SET) with less than 3 points encoded, and DISJOINT POLYLINEs with odd numbers of points encoded, are prohibited.

3. Edges of filled areas are specified to be centered on the ideal boundary of the area.

4. Control characters (codes 1-31 and 96-127) are prohibited (NUL, 0, is allowed) in graphical text strings except as required to implement permissible character set switching mechanisms.

5. The line type continuation element of MIL-D-28003 has been removed, since there is now an included Amendment 3 element which allows control of this aspect.

60.6 **Text and Fonts.** Additional fonts may be used. These fonts are similar to the basic set of 13 specified by the PostScript product of Adobe Corporation (PostScript is a registered trademark of Adobe Corporation). Equivalents of these fonts are widely available in modern publishing systems.

All font indexes and character set indexes used in the metafile must be specified by the appropriate Metafile Descriptor elements.

The case (upper/lower) of the strings in the FONT LIST element is not significant. This was not addressed in MIL-D-28003.

In forming the names of the Hershey fonts, a "/" may be used in place of the ":" for separating the components of the font name. The former is consistent with name structuring conventions in the ISO font standards.

An appendix will be prepared and added to a future revision of MIL-D-28003A to show an unambiguous mapping of character codes to glyphs for the Hershey fonts of TABLE IX.

A subset of the publicly-defined symbols of TABLE IX (useful for Technical Illustration) will be added to a future revision of MIL-D-28003A. An appendix will also be prepared and added to a future revision of MIL-D-28003A which specifies such a symbol set and unambiguously associates character codes with symbols.

The "SYMBOL" font in TABLE X contains greek characters in the familiar alphabetic positions, and various mathematical and publishing symbols in the upper code positions (159-255). In a future revision of MIL-D-28003A, the greek symbols will be given unambiguous code assignments and will be illustrated in the published text. A subset of other symbols useful for Technical Illustration (and which is widely available in publishing and engineering systems) will be added to a future revision of MIL-D-28003A. An appendix will be prepared and added to a future revision of MIL-D-28003A, which specifies such a set of symbols and unambiguously associates character codes with symbols.

The maximum allowable length of non-graphical strings (e.g., FONT LIST names, METAFILE DESCRIPTION string, etc) has been increased from 254 to 1024.

60.7 Hatch Styles and Line Types. The extended hatch styles (concrete, steel, etc) of MIL-D-28003 have been removed. The conformance requirements for them were too ill-specified, and the usefulness of them in technical illustrations for weapons systems has been challenged (they appear more appropriate to architectural and civil engineering applications) as not worth the cost. Most of these may be reproducible by the user definable hatch, available in Version 3 metafiles, which has been included.

The extended line types, by contrast, seem more widely applicable to the application area and more readily implemented. The designators for these have been changed from the private negative numbers to the newly registered positive values.

60.8 **Escapes.** All of the specified ESCAPEs of MIL-D-28003 have been removed.

1. The clear disable has side effects that have not been dealt with, and it is contrary to the intention of the ANSI/ISO 8632 standard. As a presentation directive it could accompany the metafile in some way, but should not be embedded as functionality in the metafile.

2. The use of **viewport** would be difficult to reconcile with other content types on a typical MIL-STD-1840A application tape. It should also be considered as a presentation directive that could accompany the metafile but not be embedded in it.

3. The **implicit** colour table contains reasonable requirements. However, what is achievable using it can be accomplished explicitly by other standard means. The slight, potential saving of file size does not justify defining and adding the extended element.

60.9 **Delivery Format.** MIL-D-28003 attempted to specify the delivery format on physical media of conforming basic metafiles as 80-octet records with a blocking factor specified for tape. The delivery format does need to be specified somewhere within the CALS family of standards, but MIL-D-28003 is not the right place. Such requirements are the concern of the packaging and delivery standards (e.g., MIL-STD-1840A). MIL-D-28003A has replaced the requirement with an explicit statement that MIL-D-28003A does not specify the delivery format.

60.10 **Additions--Picture Descriptor.** Following are additions of Picture Descriptor elements from Version 2 and Version 3 metafiles.

60.10.1 **SPECIFICATION MODES.** Version 3 metafiles allow two new SPECIFICATION MODES, **fractional** and **mm**, for such items as LINE WIDTH SPECIFICATION MODE. For technical illustrations (and engineering drawing as well) these are considered more precise and behave more as required by these applications than the two existing modes. In addition an INTERIOR STYLE SPECIFICATION MODE has been added to give similar control over the interiors of filled areas. These features are included in MIL-D-28003A.

60.10.2 **Grammar of Version 2 Metafiles.** The rules defining where elements can appear in the metafile are liberalized somewhat in Version 2 metafiles. For example, COLOUR TABLE can now (and should preferably) appear in the Picture Descriptor. The SPECIFICATION MODES above can also appear in the picture body. Most of these are allowed in MIL-D-28003A. MIL-D-28003A does restrict the COLOUR SELECTION MODE (also liberalized in Version 2 metafiles) to one value per picture.

60.10.3 **Prohibited Elements.** The DEVICE VIEWPORT Picture Descriptor element of Version2 metafiles is prohibited in conforming basic metafiles. In addition the bundle table setting elements are prohibited. Although the elements are not especially complicated, the utility of bundles is considered very low to this application community, and the overhead of implementing the elements does not justify the cost.

60.11 **Additions--Control.** Following are additions of Control elements from Version 2 and Version 3 metafiles.

60.11.1 **Clipping Modes.** Version 2 metafiles resolve an ambiguity in Version 1 metafiles concerning how lines and markers are clipped or cropped. Elements are added to MIL-D-28003 to set the clipping modes of lines, edges, and markers, with the Basic values constrained to a single value that makes sense in technical illustrations.

60.11.2 **Mitre Limit.** Line join is a feature of Version 3 metafiles which is included in MIL-D-28003A. The associated MITRE LIMIT element is included as well.

60.11.3 **Prohibited Elements.** Version 2 metafiles have elements to save and restore certain aspects of primitive context. These are prohibited in MIL-D-28003A.

Version 3 metafiles allow arbitrary clipping and shielding boundaries in addition to the existing rectangular clipping capability. These consist of general paths that can be used to clip and shield. While these are considered useful to technical illustration, their priority is lower than some other items. Thus they are assigned to the next revision of this specification.

Version 3 metafiles also include the ability to specify text along an arbitrary path. This is also assigned to the next revision of this specification.

60.12 **Additions--Graphical Primitives.** Following are additions of Graphical Primitive elements from Version 2 and Version 3 metafiles.

60.12.1 **Version 2 Primitives.** Version 2 added two primitives: Closed Figure, allowing the definition of a single filled area from a composite of other primitives, namely lines, arcs, polygons, etc.; and another flavor of Circular Arc, allowing seamless inclusion of arcs in Closed Figures in certain circumstances. Both primitives are adopted into MIL-D-28003A. The Closed Figure is adopted with size restrictions as the other variable size primitives have been into MIL-D-28003.

60.12.2 **Version 3 Primitives.** Parabolic arcs and hyperbolic arcs are defined in Version 3 metafiles and are included in MIL-D-28003A (thus completing the set required to translate, for example, the conic arc elements of IGES).

Cubic bezier curves are also defined, and are adopted into MIL-D-28003A (with appropriate size restrictions).

Tiled compressed raster elements are defined in a manner that allows integration of vector and raster within a picture, and in a way compatible with MIL-R-28002 and the ODA Part 7 Tiling Addendum. These are adopted into MIL-D-28003A (with appropriate size and compression option restrictions).

A "Compound Line" primitive (similar to the Version 2 Closed Figure, except not filled) is defined in Version 3 metafiles and this is adopted into MIL-D-28003A.

NURBS (non-uniform rational B-splines) are defined in Version 2 metafiles. These are included in MIL-D-28003A with appropriate restrictions on degree and size.

60.12.3 **Prohibited Elements.** The facility of externally defined symbol libraries is also defined in Version 3 metafiles. Inclusion of this facility herein is deferred until specific requirements are better defined, and such practical operational matters as naming conventions for and registration of symbol libraries have been completed.

60.13 **Additions--Primitive Attributes.** Following are additions of Primitive Attribute elements from Version 2 and Version 3 metafiles.

60.13.1 **Version Attributes.** A single new attribute, PICK IDENTIFIER, is defined in Version 2 metafiles. It is graphically meaningless, but may help to preserve certain application hooks into the graphics and so is adopted herein (it may safely be ignored by conforming basic interpreters).

60.13.2 **Definable Line Types and Hatch.** Version 3 metafiles allow user definable line (and edge) types and hatch styles. These are adopted with appropriate restrictions into MIL-D-28003A, and are considered high priority items for improving drawing quality, fidelity and translatability from other formats. Along with the user line/edge type definition elements, there are a couple of utility elements such as for specifying the continuation behavior and the initial offset in line patterns. These are adopted into MIL-D-28003A as well.

60.13.3 **Cap and Join.** Version 3 metafiles define elements to specify line cap and line join styles (and for edges as well). These are adopted into MIL-D-28003A, and are considered a fairly high priority item for improving drawing quality, fidelity and translatability from other formats.

60.13.4 **Restricted Text Controls.** Version 3 metafiles allow precise control of how text should fit the restriction box of the RESTRICTED TEXT element. This is one of the most important additions to MIL-D-28003A for controlling how text behaves, as well as for translating from other formats (e.g., IGES) having a "boxed text" model.

60.13.5 **Prohibited Elements.** Version 3 metafiles allow new, filled-interior filling methods: definable "geometric patterns" and interpolated or gradient interiors. While these are considered useful to technical illustration, they have lower priority than some other items. Therefore, they have been assigned to the next revision of this specification.

Version 3 metafiles also define attribute elements associated with external symbol libraries and arbitrary text path. As mentioned previously these features are deferred until the next revision of this specification.

60.14 **Additions--Segments.** Version 2 metafiles define the concept of "segments" to the metafile definition. A segment is specified by indicating a grouping of primitives. Once specified the group can then be instanced repeatedly into pictures, with variations on position, scaling, rotation, and inheritance of such attributes as color. This segment mechanism is adopted into MIL-D-28003A with appropriate restrictions and with the elimination of a couple of less useful elements.

60.15 **About Delimiter Elements.** The new primitives, Closed Figure, Compound Line and Tile Array, are specified by initiating and terminating a definition sequence with the Delimiter Elements BEGIN <whatever> and END <whatever>. The clipping and shielding regions of Version 3 metafiles, which are not included in MIL-D-28003A, are also specified in this way.

60.16 **About Metafile Descriptor Elements.** Version 2 metafiles define the Metafile Descriptor elements NAME PRECISION, MAXIMUM VDC EXTENT, and SEGMENT PRIORITY EXTENT. These are basic "utility" elements and are adopted into MIL-D-28003A, with restrictions where appropriate.

Version 3 metafiles define several functionally significant Metafile Descriptor elements. SYMBOL LIBRARY LIST allows the reference of external symbol libraries for inclusion of their individual symbols into the metafile. As mentioned previously, inclusion of this feature in MIL-D-28003 is deferred pending requirements study and better definition of the mechanics of the feature.

COLOUR MODEL and COLOUR CALIBRATION allow selection of color models other than RGB (Red-Green-Blue), including the CMYK color printing model. These elements also include precise calibration information relating to the source of the color. Since color is a relatively minor part of interchange of technical illustrations at this point, inclusion of these features is deferred.

FONT PROPERTIES and GLYPH MAPPING have potential to solve some of the problems in graphical interchange. Their inclusion herein is premature at this point for a number of reasons: (1) the base standards from which the elements derive, ISO 9541 and ISO 10036 have not yet completed; (2) significant collections of glyphs have not yet been registered; (3) the requirements of CALS for symbols and glyphs in vector graphics have not yet been defined; and (4) the font substitution capabilities of FONT PROPERTIES can not be used effectively until further liberalization of font usage herein is defined and certain companion changes are implemented in MIL-STD-1840A.

APPENDIX B

LOOKING BEYOND REVISION A OF MIL-D-28003

10. **SCOPE.**

10.1 **Scope.** This appendix is provided for informative purposes, to assist in evaluating Revision A of MIL-D-28003. This appendix is not a mandatory part of this specification. It provides a rationale for why certain features have been deferred from Revision A processing, and offers suggestions for how these features can be processed in future as either minor amendments or part of the next full revision cycle. The material contained herein is intended for guidance only.

20. **APPLICABLE DOCUMENTS.** This section is not applicable to this appendix.

30. **OVERVIEW.** Appendix A focusses on the differences between MIL-D-28003 and MIL-D-28003A, discussing all of the new features of Version 2 and Version 3 metafiles. Some features of these versions are of no interest to the technical publishing and engineering communities of CALS. Other features are useful but are deferred for one of two reasons: adequate groundwork and requirements definition does not yet exist; or the feature, although useful, is not a high-enough priority when both the added implementation burden and the number of other features already being added are considered.

The content of future revisions to this specification, as well as the best means of making and timing the additions, are now being considered.

The bulk of the changes will probably constitute a significant revision, which for the remainder of this discussion will be designated MIL-D-28003B. There are some items that ideally should have been included herein. However, these items have been adequately prepared and should not delay the start of the review process of this specification. In addition the review and reconciliation process for MIL-D-28003A represents a valuable opportunity for soliciting comments on requirements.

MIL-D-28003B probably should start its review process about 2 years after MIL-D-28003A is published. It is possible that some items not presently included in MIL-D-28003A, and upon which requirements comments are solicited, can be processed in the interim by minor amendment. For the purposes of this discussion these are referred to as "Interim Items."

40. **INTERIM ITEMS.** Need has repeatedly been expressed for a "symbol font," containing various common publishing, mathematical, and engineering symbols which may need to occur as an integral part of a graphical text string. The exact requirement has not yet been defined. There are a number of candidate symbol sets that can provide source material once the requirements are determined:

1. public domain Hershey symbol sets;

2. symbols specified for use with IGES;

3. proprietary but common sets such as the Adobe PostScript "Symbol" font (PostScript is a registered trademark of Adobe Corporation);

4. the registries of technical societies.

Continued examination of the symbol problem will result in a solution or recommendation that will be added to MIL-D-28003A without a major revision process.

50. **SUBSTANTIAL ADDITIONS--28003B.**

50.1 **More Functionality from Version 2 and 3.** There are features of Version 2 metafiles which have not been adopted into MIL-D-28003A. These are not targeted for adoption at any time. These features include:

1. settable bundles;

2. device viewport controls;

3. save/restore context;

4. a couple of implementation dependent segment attributes.

Some features of Version 3 metafiles have not been adopted into MIL-D-28003A. These features are targeted for adoption into the next major revision (MIL-D-28003B) of this specification. The need for certain other features is uncertain at this point. They include the following:

1. **NURBS.** These curves are becoming more common in engineering systems; adoption into MIL-D-28003B is probable.

2. **SYMBOLS.** Access to external libraries of standardized symbols is potentially very useful to engineering drawing and technical illustration; adoption into MIL-D-28003B is probable after mechanics of the elements are worked out.

3. **FONT PROPERTIES.** The ability to specify acceptable substitutes for a larger class of allowed but non-basic fonts is one goal of additional font improvements in MIL-D-28003B; adoption into MIL-D-28003B is probable.

4. **GLYPH MAPPING.** Access to large families of registered typographic and technical glyphs is desirable; adoption into MIL-D-28003B is probable after the glyph registry develops somewhat.

5. **GEOMETRIC PATTERN** and **INTERPOLATE INTERIOR.** While these features are essential to the highest quality results in graphic arts and presentation graphics, the precise requirements in Technical Illustrations are uncertain (and in engineering drawing are low); adoption or limited adoption into MIL-D-28003B is probable, pending further study of requirements.

6. **GENERALIZED TEXT PATH.** While this feature is essential to the highest quality results in graphic arts and presentation graphics, and is anticipated to be widely used in cartography, the precise requirements in Technical Illustrations are uncertain; adoption into MIL-D-28003B is possible pending further study of requirements.

7. **ARBITRARY CLIP/SHIELD.** The same as for GEOMETRIC PATTERN (they are functionally equivalent if the question of nesting is discounted).

8. **COLOR.** Because the great majority of Technical Illustration currently is monochrome, a strong requirement has not been generated for advanced color models and color calibration capabilities within the CALS community; adoption into MIL-D-28003B is unlikely unless additional requirements are generated. However, the increasing use of "softcopy" (CRTs) for document

presentation implies that some additional requirements work is needed on this question.

50.2 **More Fonts.** MIL-D-28003A has increased the selection of fonts which can be used by allowing a set which is widely available but not public domain. The additional fonts are of higher quality than the public domain Hershey fonts. MIL-D-28003B will likely expand the set of allowable fonts further. This will likely be in conjunction with some of the font substitution mechanisms just described, and ideally also in conjunction with a font resource content type (see next section).

50.3 **Font Resource Content Type for MIL-STD-1840A.** A major issue is achieving modern, high-quality typography in an open interchange environment such as CALS while still preserving universal printability. One way to solve font interoperability problems is for the font resource, or minimally parts of the font resource, to be delivered with the graphics and text of an electronic document. If the font metrics are available for the fonts utilized both in the graphics and in the text layout and formatting, then the approximation of unavailable fonts can be improved by an order of magnitude. If the shape descriptions themselves are also delivered, then substitution and approximation is no longer an issue.

There are ISO projects in reasonably-advanced stages that standardize the basic technology for interchange of font resources. These are based on widespread commercial practice.

Within the next two years the MIL-STD-1840A standard should be modified to allow "Font Resource" to be one of the allowable content types delivered on tape or whatever medium is being used. This is a necessary adjunct to solutions for presentation of the revisable content types of MIL-STD-1840A, for both graphical font presentation and text presentation and formatting.

Results of adding a Font Resource content type would be integrated into MIL-D-28003B.

50.4 **Character Set Problems.** Further study is needed on the issues of coordination of character, symbol, and glyph sets between the revisable content types of MIL-STD-1840A. MIL-D-28003B will have better basic capabilities in the areas of character sets and typographical symbols, and ideally these will be better coordinated with the other content types.

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

Preparing Activity
OSD-CL
(Project ILSS - 0044)

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

APPENDIX 2

CGM AMENDMENT 3 TEXT

ISO/IEC 8632-1:1987/Am.3:1991

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer of Picture Description Information

Part 1

Functional Specification

CONTENTS

0 Introduction 1

2 References 2

3 Definitions and abbreviations 3

4 Concepts 5

5 Elements 49

6 Defaults 84

7 Annex D 86

8 Annex I 87

9 Annex J 116

0 Introduction

Page 3

Subclause 0.8, add the following new text after the first paragraph:

ISO/IEC 8632-1:1987/Am.3:1990 uses font concepts and the font architecture defined in ISO/IEC 9541-1 (currently at DIS) for defining CGM references to fonts and font resources. The font properties of ISO 9541-1 are adopted where appropriate to define CGM mechanisms to provide information useful for font substitution between parties interchanging Metafiles. ISO 8632-1:1987/Am.3:1990 includes from 9541-1 the minimum font description subset defined in ISO 9541-2. Clause 3 contains a number of glossary definitions that are taken from and are identical to those found in ISO 9541-1.

ISO 8632-1:1987/Am.3:1990 uses a colorimetrically precise reference colour space to allow for interchange of precise colour specifications. ISO 8632-1:1987/Am.3:1990 uses concepts defined in ISO/IEC 8613/Ad.2 (currently at Committee Draft) which are based on CIE publications. ISO 8613/Ad.2 provides tutorial material on relevant definitions and colour concepts, which is useful for understanding the material in this Standard but is not incorporated into this Standard in that detail.

Page 3

Subclause 0.9, change:

The annexes do not form an integral part of this part of ISO 8632 but are included for information only.

to:

The annexes are either an integral part of this part of ISO 8632 (normative) or included for information only (informative), as indicated at the start of each annex.

Page 3

Insert new subclause 0.10:

0.10 Amendments and versions

This International Standard defines a version 3 Computer Graphics Metafile. Version 1 Metafiles conforming to ISO 8632 are defined by ISO/IEC 8632:1987. Version 2 Metafiles are defined by the amendment ISO/IEC 8632:1987/Am.1:1990. Version 3 Metafiles are defined by amendment to version 2 of CGM, which amendment is defined by the ISO/IEC 8632:1987/Am.3:1991. A valid version 1 Metafile is both a valid version 2 Metafile and a valid version 3 Metafile. A valid version 2 Metafile is a valid version 3 Metafile.

2 References

Page 5

Clause 2, References add the following references:

CIE Publication 17-4 International Lighting Vocabulary, 1987 (4th edition)

CIE Publication 15-2 Colorimetry, 1986 (2nd Edition)

CIE Publication S002, Colorimetric Observers, 1986 (1st edition)

ISO/IEC 10036:1988, Information processing systems — Procedure for registration of glyph and glyph collection identifiers.

ISO/IEC 9541-1:1990, Information processing systems — Font information interchange, Part 1: Architecture.

ISO/IEC 8613:1989, Information processing systems — Text and office systems — Office document architecture (ODA) and interchange format — Ad.2 to add colour capabilities.

SMPTE, Recommendeed Practice RP145, Colour Monitor Colorimetry, 1986.

ISO 2846, Set of printing inks for offset printing — colorimetric characteristics, 1975.

ISO/IEC 8613-7:1989, Information processing systems — Text and office systems — Office document architecture (ODA) and interchange format — Part 7: Raster Graphics Content Architecture, Draft of a Tiled Raster Graphics Addendum, 1990.

J. Ziv and A. Lempel, A Universal Algorithm for Sequential Data Compression, IEEE Transaction of Information Theory, Vol. IT-23, No. 3, May 1977, pp. 337-343.

J. Ziv and A. Lempel, Compression of Individual Sequences via Variable-Rate Coding, IEEE Transaction of Information Theory, Vol. IT-24, No. 5, Sept. 1978, pp. 530-536.

T.A. Welch, A Technique for High-Performance Data Compression, IEEE Computer, Vol. 17, No. 6, June 1984, pp. 8-19.

3 Definitions and abbreviations

Page 6

Sub-clause 3.1, add or change the following definitions:

3.1.1 additive colour mixing: Superposition or other nondestructive combination of lights of different chromaticities.

3.1.2 brightness: Attribute of a visual sensation according to which an area appears to emit more or less light.

3.1.3 chroma: The colourfulness of an area judged in proportion to the brightness of a similarly illuminated area that appears white or highly transmitting.

3.1.4 chromaticity: Ratio of each of a set of three tristimulus coordinate values to their sum. As the sum of the three chromaticity coordinates equals 1, two of them are sufficient to define a chromaticity.

3.1.5 CIELAB: A CIE recommended, approximately uniform colour space with rectangular coordinates L^*, a^*, and b^*. L^* is the approximate correlate of lightness, a^* and b^* are used to calculate an approximate correlate of chroma. CIELAB uses the tristimulus ratios X/X_n, Y/Y_n, Z/Z_n instead of chromaticity coordinates as in CIELUV. X_n, Y_n, Z_n are the values of X,Y,Z for the appropriate chosen reference white. The reduced perceptual significance of a given difference in chromaticity as the colour becomes darker is incorporated by using the tristimulus ratios.

3.1.6 CIELUV colour space: A CIE recommended, approximately uniform colour space with rectangular coordinates L^*, u^*, and v^*. L^* is the approximate correlate of lightness, u^* and v^* are used to calculate an approximate correlate of chroma. The colour stimulus is described by Y,u,v and the the reference white by Y_n, u_n, v_n. A given difference in chromaticity is reduced in magnitude by the factor L^* as the colour becomes darker,

3.1.7 CIE tristimulus values: Amounts of the three reference colour (of a colour stimulus) stimuli, in a given trichromatic system, required to match the colour of the stimulus considered.

3.1.8 CIE uniform colour spaces: Two CIE recommended, approximately uniform colour spaces, CIELAB and CIELUV, are allowed in the CGM. These colour spaces are non-linear transformations of the CIE 1931 XYZ tristimulus space, into the approximate perceptual correlates of lightness and chroma. CIELAB and CIELUV closely approximate uniform colour spaces over small distances, and they each provide an approximately uniform measure of perceived colour differences. Both colour spaces allow for the perceptual effect that a given difference in chromaticity represents a smaller and smaller colour difference as the lightness is reduced.

3.1.9 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).

3.1.10 colour component: One of the dimensions of a colour space.

3.1.11 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, CIELUV, or CYMK.

3.1.12 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.

3.1.13 colour space: See COLOUR MODEL.

3.1.14 colour stimulus: Visible radiation entering the eye and producing a sensation of colour.

3.1.15 colour value: Value of the n-tuple of components describing a colour in a given colour model.

3.1.16 escapement: During the rendering of text strings onto a display, the movement of the current position on the display surface after a glyph representation is imaged.

3.1.17 escapement point: A glyph metric; a point in the glyph coordinate system, to which the current position on the display surface is usually translated, after the glyph representation is imaged.

3.1.18 font: A collection of glyph images having the same basic design, e.g., *Courier Bold Oblique.*

3.1.19 font family: A collection of fonts of common design, e.g., *Courier, Courier Bold, Courier Bold Oblique.*

3.1.20 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.

3.1.21 glyph: An identified abstract graphical symbol independent of any actual image.

3.1.22 illuminant: Radiation with relative spectral power distribution defined over the wavelength range that influences object colour perception.

3.1.23 intensity, luminous: The light flux per unit solid angle.

3.1.24 luminance: In a given direction, at a point in the path of a beam, the luminous intensity per unit projected area.

3.1.25 luminance factor: Ratio of the luminance to that of the perfect reflecting or transmitting diffuser identically illuminted.

3.1.26 perfect diffuser: An ideal isotropic diffuser with reflectance (or transmittance) equal to unity.

3.1.27 posture: The extent to which the shape of a glyph or set of glyphs appear to incline, including any consequent design or form change.

3.1.28 primary colour stimuli: Three selected coloured lights used to specify the colour of any light presented by the amounts of the three lights that must be mixed additively to produce light matching the light presented. (Any three coloured lights may serve as primaries provided no one of them can be matched by a mixture of the other two. To achieve the maximum gamut of colours by additive mixture, saturated red, green, and blue primaries are commonly used.)

3.1.29 primary colourants: A small number of clolourants (dyes or pigments) that may be mixed subtractively to produce a large gamut of colours. (The most common primary colourants are yellow, magenta (purplish red), and cyan (greenish blue).)

3.1.30 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.

3.1.31 reference colour model: Basic colour model within CGM relative to which relationships to specifiable colour models (RGB, CYMK, CIELUV, and CIELAB) are calibrated. The reference colour model is defined by the CIE 1931 standard colorimetric system (XYZ).

3.1.32 saturation: The colourfulness of an area judged in proportion to its brightness.

3.1.33 subtractive colour mixture: Mixing of absorbing media or superposition of filters so that the spectral composition of light passing through the combination is determined by simultaneous or successive absorption. (This kind of colour mixing occurs in hardcopy images and in colour photography.)

3.1.34 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.

3.1.35 trichromatic system: System for specifying colour stimuli in terms of tristimulus values, based on matching colours by additive mixture of three suitable chosen reference colour stimuli.

3.1.36 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 PROTECTION REGION and END PROTECTION REGION elements.

A tile array may be defined by tile array elements occurring between BEGIN TILE ARRAY and END TILE ARRAY.

The exact list of elements which may occur in any of these definition states will be found in the State Table, Table 3.c.

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 COMPOUND PATH
END COMPOUND PATH
BEGIN PROTECTION REGION
END PROTECTION REGION
BEGIN TILE ARRAY
END TILE ARRAY
COLOUR MODEL
COLOUR CALIBRATION
FONT PROPERTIES
GLYPH MAPPING
SYMBOL LIBRARY LIST
INTERIOR STYLE SPECIFICATION MODE
PROTECTION REGION INDICATOR
GENERALIZED TEXT PATH MODE
MITRE LIMIT
HYPERBOLIC ARC
PARABOLIC ARC

NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER
SYMBOL
BITONAL TILE
TILE
LINE & EDGE TYPE DEFINITION
LINE CAP
LINE JOIN
LINE TYPE CONTINUATION
LINE TYPE INITIAL OFFSET
TEXT SCORE TYPE
RESTRICTED TEXT TYPE
HATCH STYLE DEFINITION
GEOMETRIC PATTERN
INTERPOLATED INTERIOR DEFINITION
EDGE CAP
EDGE JOIN
EDGE TYPE CONTINUATION
EDGE TYPE INITIAL OFFSET
SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL SIZE
SYMBOL ORIENTATION
SEGMENT VISIBILITY

Page 11

Add the following after subclause 4.3.3:

4.3.4 Font List and Font Resources

ISO 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. This concept is described in ISO/IEC 9541-1, Annex B.

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 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

Concepts

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 those in the Minimum Font Description Subset of ISO 9541. Applications may use private or registered extensions to access additional properties from amongst the 9541 font properties.

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 necessary 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 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 (Am.1, page 5, paragraph 2)

Subclause 4.4.2, first line, change:

direct (RGB) colour

to

direct colour

Page 14

Subclause 4.4.3, first line, change:

Line width, marker size, and edge width may be specified in more than one way. The width of lines, for example, may be specified as either a measure in VDC units or as a scale factor to be applied to a device-dependent nominal line width at interpretation time.

to:

Line width, marker size, interiors of filled-area elements, and edge width may be specified in more than one way. The width of lines, for example, may be specified as either a measure in VDC units, a scale factor to be applied to a device-dependent nominal line width at interpretation time, a fraction of the device view surface, or a measure in millimetres.

Page 14

Subclause 4.4.6, second paragraph, first line, change:

RGB

to

a direct colour

Page 15

After subclause 4.5.3, add:

4.5.4 Compound Clipping and Shielding

The clipping and shielding elements consist of BEGIN PROTECTION REGION, END PROTECTION REGION, and PROTECTION REGION INDICATOR. The BEGIN PROTECTION REGION and END PROTECTION REGION elements are delimiter elements and the PROTECTION REGION INDICATOR element is a control element.

For an individual protection region the inside of the protection area is defined in the same way that the interior of a filled-area element is defined (4.6.4.4).

When a protection region is set for clipping only the portions of the graphic elements inside or on the boundary of the protection region are drawn. When a protection region is set for shielding only the portions of the graphic elements outside the protection region are drawn.

Multiple clip and shield regions may be in effect simultaneously. If shield regions overlap, the effective shield region is the union of the individual shield regions. If clip regions overlap, the effective clip region is the intersection of the individual clip regions.

4.5.5 Mitre Limit

The control elements include an element for controlling how lines and filled-area edges are drawn. Line or edge joins may be rendered with mitre join style, as determined by the appropriate line and edge attribute elements. The mitre join is formed by projecting the outer edges of the lines or edges at the corner until the projections meet at a point. When mitre joins are being rendered, there is the possibility of the mitre projecting very far if the line segments meet at a very sharp angle at the vertex. The MITRE LIMIT provides a means of specifying that long mitres are to be truncated at some point to form a flat bevel. *Mitre length* is defined to be the distance from the point at which the inside edges of adjoining line segments meet to the point at which the outside edges meet. The parameter of MITRE LIMIT is a single real number. If the mitre length divided by the line width exceeds the value of the parameter of the MITRE LIMIT element, then the mitre join is truncated at that limit.

Page 15

Subclause 4.6, add the following to the list of graphical primitive elements:

HYPERBOLIC ARC
PARABOLIC ARC
BITONAL TILE

Concepts

TILE
NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER
SYMBOL

Page 15

Subclause 4.6, add the following paragraph after the list of graphical primitive elements:

In addition, a tile array compound graphical primitive may be defined by a sequence of TILE and BITONAL TILE elements between the BEGIN/END TILE ARRAY delimiters.

Page 15

Subclause 4.6 add the following to the list of line elements:

HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER

Page 16

Subclause 4.6, before the last paragraph add:

The tile array primitive elements are:

BITONAL TILE
TILE

The single symbol primitive element is:

SYMBOL

Page 16 (Am.1, page 9, table 1a and text before table 1a)

Subclause 4.6, first paragraph, second sentence, change:

'compound text' and 'closed figures'

to:

'compound text', 'closed figure' and 'compound line'

Page 16 (Am.1, page 9, table 1a)

Subclause 4.6, table 1a, add:

Compound Primitive	First Element	Primitives Included	Other Elements	Final Element
Compound Line	BEGIN COMPOUND PATH (Note 6)	Line primitives GDP (Note 5)		END COMPOUND PATH

6 For path type 'compound line'.

Page 16 (Am.1, page 9, table 1a)

Subclause 4.6, table 1a, change NOTE 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, ...

to:

Whether a GDP may contribute to compound text, closed figure or compound line, and whether or how it specifies that the text open state, figure open state or path state be opened, ...

Page 16

Subclause 4.6.1.1, add to the end of the subclause:

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 5.6.22, and the principles underlying the transformable parameterization are described in 4.6.8.

PARABOLIC ARC: generates a parabolic arc; the parameterization is described in 5.6.23, and the principles underlying the transformable parameterization are described in 4.6.9.

NON-UNIFORM B-SPLINE: generates a Non-Uniform B-Spline curve; the parameterization is described in 5.6.24, and the principles underlying the definition of the element are described in 4.6.10.1.

NON-UNIFORM RATIONAL B-SPLINE: generates a Non-Uniform Rational B-Spline curve; the parameterization is described in 5.6.25, and the principles underlying the definition of the element are described in 4.6.10.1.

POLYBEZIER: generates a sequence of one or more cubic Bezier curves; the parameterization is described in 5.6.26 and the principles underlying the definition of the element are described in 4.6.10.2.

Subclause 4.6.1.3, last sentence, change:

The ARC primitives

to:

The arc primitives (circular, elliptical, hyperbolic, and parabolic) and spline primitives (Bezier and Non-uniform B-splines)

Concepts

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

Subclause 4.6.3.3, add the following text at the end of the subclause:

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 COMPOUND PATH and 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 are not rotated relative to the text path. When GENERALIZED TEXT PATH MODE is 'axis-tangential' the characters are positioned along the text path and the character orientation vectors for each character are rotated by an amount equal to the angle of the tangent to the text path at the character position.

GENERALIZED TEXT PATH MODE is a Control Element. Illustrations of GENERALIZED TEXT PATH MODE are shown in figures 11b, 11c, 11d, 11e, 11f, 11g, and 11h.

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 Tile Array Elements.

BITONAL TILE:	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 cell 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 TILE element.
TILE:	defines a rectangular raster image, either uncompressed or compressed according to one of a number of compression methods. The colours associated with the cells 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.

Tile Array — A tile array is a compound raster image primitive, whose definition is delimited by the BEGIN TILE ARRAY and END TILE ARRAY delimiter elements. Between the delimiter elements is a series of equally sized individual "tiles" which form a contiguous rectangular block. Each tile is defined by a TILE or BITONAL TILE element. The first tile — the tile with tile identifier 1 — is placed at the ***position*** parameter of the BEGIN TILE ARRAY element. Any subsequent tiles are placed at the tile position corresponding to their ***tile identifier*** parameter. The tile positions are numbered as shown in figure X.

The TILE and BITONAL TILE elements are similar to CONNECTING EDGE, see 4.6.1.1, in that they are not independent graphical primitives in the same sense as POLYLINE and CELL ARRAY. They contain no positioning or dimensioning information. They contain only the raster content of a single raster tile and any control parameters which apply to individual tiles. The complete Tile Array primitive is formed by one or more tiles between BEGIN TILE ARRAY and END TILE ARRAY. BEGIN TILE ARRAY contains all parameters which apply to the collection of tiles (if there is more than one) and uniformly to each tile in the collection.

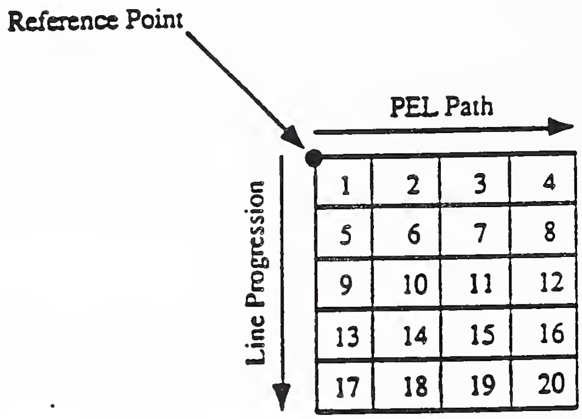

Figure 1a. Ordering and layout of tiles by index

4.6.5.1.1 Relationship to CELL ARRAY. Both tile arrays and cell arrays are composed of cells. While cells in cell arrays are subject to all transformations, cells in tiles are always axis aligned and rectangular. They scale to the display surface but do not otherwise transform.

4.6.5.1.2 Allowable states for tiles and tile array elements. The tile elements may appear only in Tile Array State (TAS). Tile Array compound elements, delimited by BEGIN TILE ARRAY and END TILE ARRAY, may appear only in Picture Open State. They may not appear in segments or other compound primitive definitions.

4.6.5.1.3 Compressed cell data. The cell colour data of the tile elements is a compressed stream of cell colour specifiers. The datatype is Bitstream. For the BITONAL TILE element 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) define a technique for representing and encoding the compressed binary data object.

4.6.5.1.4 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 tile array is initiated by the BEGIN TILE ARRAY delimiter element and terminated by the END TILE ARRAY element. During tile array definition subsequent tile elements define individual tiles within the tiled image. The number of tiles is determined by the parameters of the BEGIN TILE ARRAY element. A tile array contains one or more tiles.

Concepts

The number of tiles defined during tile array definition must match the number indicated by the BEGIN TILE 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 the tile space, relative to the *position* parameter of the BEGIN TILE ARRAY element. There may be a border of undefined data surrounding the actual image. The data is undefined in the sense that it is not part of the actual image and should not be drawn by the interpreter. This undefined data is however included in the encoding of those tiles which overlap the border and the defined image — the number of cells encoded in the Bitstream parameter of each tile is the same for these "border" tiles as for tiles which lie entirely within the defined image. Therefore the border cells must have some value assigned — it is typical to set them all identically to the background colour.

NOTE — The offset and size parameters are provided because typically there may be parts of the tile array which contain no useful information and are simply artifacts of tiling. For example, the actual raster size of a scanned image may not be an integral multiple of convenient tile sizes.

Page 19

Subclause 4.6.7.1, add to the end of the subclause:

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 ellipse that is centred 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 centred 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 centred at the origin. M maps the unit vectors $\mathbf{u}_1=(1,0)$ and $\mathbf{u}_2=(0,1)$ respectively onto P_1 and P_2. These principles generalize easily to ellipses which are not centred at the origin — there is a translation term in the mapping so that the transformation is not linear but is affine.

Page 19 (Am.1, page 11)

Subclause 4.6.8, add the following after the subclause:

4.6.9 Hyperbolic Arc Element

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 point $\mathbf{u}_1$ and at $\mathbf{u}_1$ the tangent to the hyperbola is parallel to $\mathbf{u}_2$; the hyperbola has "centre" (the point where the asymptotes cross) at the origin. Then for any non-degenerate hyperbola "centred" at the origin there is a linear transformation which maps the canonical hyperbola onto the given hyperbola. This transformation maps $\mathbf{u}_1$ and $\mathbf{u}_2$ 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 $P_1 - P_2$. A pair of points with such properties is referred to as a *conjugate radius endpoint/transverse radius endpoint* pair. The *transverse radius* endpoint is the one which lies on the hyperbola; the *conjugate radius* endpoint does not. These points (plus the centre point) 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 then 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 centre 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 centre.

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.

4.6.10 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 $2(x + y) = (x-y)^2 + 1$ for $x \leq 1$ and $y \leq 1$. This parabolic arc is symmetric about the line $y=x$, has endpointss $\mathbf{u}_1$ and $\mathbf{u}_2$, and passes through the fourth quadrant, the origin, and the second quadrant between $\mathbf{u}_1$ and $\mathbf{u}_2$.

The general parabolic arc is parameterized by the endpoints of the arc, $\mathbf{P}_1$ and $\mathbf{P}_2$ and the intersection of the tangents to the arc at the endpoints. This intersection point is called the "centre" of the parabolic arc, $\mathbf{C}$. Define $\mathbf{V}_1 = \mathbf{P}_1 - \mathbf{C}$ and $\mathbf{V}_2 = \mathbf{P}_2 - \mathbf{C}$, and form the 2x3 matrix $\mathbf{M}$ whose first column consists of the components of $\mathbf{V}_1$, second column consists of the components of $\mathbf{V}_2$, and third column consists of the components of $\mathbf{C}$. For non-degenerate parabolic arcs $\mathbf{M}$ is an affine transformation that maps points on the canonical parabolic arc onto points on the given parameterized parabolic arc.

4.6.11 Spline Curve Elements

The CGM provides three spline curve elements: non-uniform B-splines; non-uniform rational B-splines; and Bezier curves.

4.6.11.1 Non-uniform B-splines

The CGM provides both rational and non-rational B-splines of varying orders.

4.6.11.1.1 Parameterization. The non-uniform B-spline is parameterized by a spline order, a list of knots, a list of control points, and parameter range limits defining the curve section to be drawn.

The non-uniform rational B-spline is parameterized by a spline order, a list of knots, a list of control points, a list of weights associated with the control points, and parameter range limits defining the curve section to be drawn.

4.6.11.1.2 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) for B-spline or 3D(x,y,w) for rational B-spline)
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:

$$B_i^0(t,[T_i,T_{i+1}]) = \begin{cases} 1 & \text{if } T_i \le t \le T_{i+1} \\ 0 & \textit{otherwise} \end{cases}$$

$$B_i^k(t,[T_{i-k},\ldots,T_{i+1}]) = \frac{(t - T_{i-k}) * B_{i-1}^{k-1}(t,[T_{i-k},\ldots,T_i])}{T_i - T_{i-k}} + \frac{(T_{i+1} - t) * B_i^{k-1}(t,[T_{i-k+1},\ldots,T_{i+1}])}{T_{i+1} - T_{i-k+1}}$$

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.

4.6.11.2 Polybezier.

This element defines one or more cubic Bezier curves.

4.6.11.2.1 Parameterization. The polybezier is parameterized by a list of points and an indicator specifying the degree of continuity between the individual Bezier curves. If there n Bezier curves then the data list will contain:

— 4n points if the continuity parameter is ***discontinuous***;

— 3n+1 points if the continuity parameter is ***continuous***;

— 2n+2 points if the continuity parameter is ***smooth***;

In the case of ***discontinuous*** the point list is divided into consecutive sets of 4 points. Each set defines a single Bezier curve as defined below.

In the case of ***continuous*** then after the first set of 4 points, defining the first Bezier curve, the subsequent curves are defined by three points each. The first point of each curve definition is omitted because it is identical to the last point of the preceding definition.

In the case of ***smooth*** then after the first set of 4 points, defining the first Bezier curve, the subsequent curves are defined by two points each. The first two points of each curve definition are omitted because their contribution to the curve definition can be derived from the last two points of the preceding curve definition.

4.6.11.2.2 Geometric concepts. The following discussion assumes that there are 4 points defined for each Bezier curve. In one of the continuity conditions described above, one or two of the points may not actually be in the parameter data of the element. 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 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 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

$$X_1 = X_0 + \frac{C_x}{3}$$

$$Y_1 = Y_0 + \frac{C_y}{3}$$

$$X_2 = X_1 + \frac{(C_x + B_x)}{3}$$

$$Y_2 = Y_1 + \frac{(C_y + B_y)}{3}$$

$$X_3 = X_0 + C_x + B_x + A_x$$

$$Y_3 = Y_0 + C_y + B_y + A_y$$

Page 20

Subclause 4.6, add the following new subclause:

4.6.12 Symbol Elements

4.6.12.1 Description

This Standard defines mechanisms to access external symbol libraries and include their symbols in the metafile by reference. There is one symbol primitive 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.

4.6.12.2 Attributes

The selection, sizing and placement of symbols is specified by the attribute elements SYMBOL SIZE, SYMBOL COLOUR, SYMBOL ORIENTATION, and SYMBOL LIBRARY INDEX.

Selection of the current symbol library from the list of available libraries is specified 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 access 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 11i. The symbol extent box is the design size of the symbol. It is used to define the transformation to scale the symbol to the size specified in SYMBOL SIZE. The symbol need not be entirely contained within the symbol extent box. Each symbol has 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 SIZE specifies the VDC sizes to which the design height of the symbol (the design distance between the top and the bottom of the symbol extent box) and the design width of the symbol (the design distance between the left and right side of the symbol extent box) shall be scaled for symbol display.

SYMBOL ORIENTATION specifies a symbol up vector and base vector, which define the orientation, skew, and distortion of the symbol.

4.6.12.3 Usage

The way in which software invoking the metafile generator and/or the metafile generator itself may use SYMBOL ORIENTATION is described. To generate the SYMBOL ORIENTATION and SYMBOL SIZE 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 whose length is the symbol width and whose direction is negative 90 degrees from the up vector. This pair of vectors may be transformed before being passed to the metafile generator to generate the SYMBOL ORIENTATION and SYMBOL SIZE elements. If the resultant vectors are not orthogonal, the symbol extent box becomes a parallelogram, and the symbol is skewed. If the positive angle from the up vector to the base vector is less than 180°, the symbol is mirror imaged. The height and width parameters of the SYMBOL SIZE element are derived from the length and width of the transformed vectors, and the indicator as to how the interpreter is to scale the symbol is generated from the application requirements.

The SYMBOL SIZE and SYMBOL ORIENTATION are decoupled. Thus, to a metafile interpreter, the lengths of the vectors in SYMBOL ORIENTATION are not significant; only their directions are significant. The SYMBOL SIZE allows the generator to request that the symbol height is to be scaled without distortion of the symbol aspect ratio, or the symbol width is to be scaled without distortion of the aspect ratio, or both are to be scaled with possible distortion of the aspect ratio.

Page 21

Subclause 4.7, Table 1, add the following elements to the list of individual attribute elements:

LINE AND EDGE TYPE DEFINITION
LINE CAP
LINE JOIN
LINE TYPE CONTINUATION
LINE TYPE INITIAL OFFSET
TEXT SCORE TYPE
RESTRICTED TEXT TYPE
HATCH STYLE DEFINITION
GEOMETRIC PATTERN
INTERPOLATED INTERIOR DEFINITION
EDGE CAP
EDGE JOIN
EDGE TYPE CONTINUATION
EDGE TYPE INITIAL OFFSET
SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL SIZE
SYMBOL ORIENTATION

Page 22

Subclause 4.7, Table 2, add the following elements to the Affected Primitives of LINE elements:

HYPERBOLIC ARC
PARABOLIC ARC
NON-UNIFORM B-SPLINE
NON-UNIFORM RATIONAL B-SPLINE
POLYBEZIER

Page 23

4.7.4.1 a), add to the list of interior styles:

geometric pattern, interpolated

Page 23

4.7.4.1 d), add to the end of the sentence:

interior style is selected.

Page 24

Replace section title of 4.7.5 and the entire paragraph describing the effects of *absolute* and *scaled* specification modes (line, marker, and edge) with:

4.7.5 Specification modes and transformation of aspects

The CGM provides the mechanism for selecting different modes by which geometric information related to line width, line type, edge width, edge type, marker size, and fill interiors is specified. The following specification modes are defined for aspects relating to size and distance:

absolute: specification units are VDC;

scaled: specification units are a scale factor to be applied by the interpreter to a device-dependent "nominal" measure;

fractional: specification units are interpreted as a fraction of the horizontal dimension of the default device viewport;

mm: specification units are millimetres.

Some primitives (those in segments) may have a transformation associated with their VDC definition. The application of this transformation gives the actual appearance of the primitive in VDC. Also all primitives in a picture may be subject to a transformation of VDC to the display device, which defines their final displayed size and appearance. This transformation may be explicitly specified by the DEVICE VIEWPORT element or partially specified by the SCALING MODE element. Mode *absolute* means that the affected aspects (e.g., line width, line caps and joins, line dash and gap lengths, etc) are subject to any and all transformations. In the other three modes none of these aspects are subject to any transformations.

In mode *absolute* the interpreter conceptually renders the associated aspects in VDC space before it applies any associated transformations. For the other three modes the aspect is conceptually rendered after all transformations have been applied to the geometry of the primitive, in the drawing space of the device.

The three non-transformable modes are distinguished by these properties:

- *scaled* gives a result which is completely device and interpreter dependent — neither the final displayed sizes nor their relationship to the rest of the displayed picture are precisely controllable;
- *fractional* gives a precisely controllable way of specifying the sizes, according to their relationship to the rest of the picture but not in terms of actual physical measurements; it is device independent in giving a picture whose components maintain a fixed relationship to each other at all display sizes, but it does not provide for invariant actual sizes across a range of picture sizes.
- *mm* gives a precisely controllable way of specifying the sizes in terms of their actual physical measurements which remain invariant as display size varies; it is device dependent in that the request may not make sense at some display sizes on some devices.

Concepts

Page 15 (Am.1, page 7, paragraph 6)

Subclause 4.5.2, change:

When a width or size specification mode is 'scaled'...

to:

When a width or size specification mode is *scaled*, *fractional*, or *mm*...

Page 16 (Am.1, page 10, paragraph 5, first sentence)

Subclause 4.6.1.3, change:

If the line width is measured in VDC units it is subject to...

to:

If the LINE WIDTH SPECIFICATION MODE is *absolute* then all line aspects — line width, line cap, line join, line dash and gap lengths — are subject to...

Page 16 (Am.1, page 10, paragraph 5, last sentence)

Subclause 4.6.1.3, change:

If the line width is specified as a scale factor...

to:

If the line width is specified in mode *scaled*, *fractional*, or *mm*...

Page 17 (Am.1, page 11, line 1)

Subclause 4.6.2.3, change:

If the marker size is specified as a scale factor...

to:

If the marker size is specified in mode *scaled*, *fractional*, or *mm*...

Page 19 (Am.1, page 11, paragraph 2 of 4.6.4.6)

Subclause 4.6.4.6, change:

The vectors of PATTERN SIZE are subject to all transformations.

to:

If the INTERIOR STYLE SPECIFICATION MODE is *absolute* the geometric aspects of fill interiors are subject to all transformations. These apects include PATTERN SIZE; direction, spacing, and width of hatch lines; and reference geometry of interpolated interior. If the mode is *scaled*, *fractional*, or *mm*, none

Concepts

NOTE — If anisotropic transformations are in effect then the height of the text will change for paths other than straight lines.

Page 37

Subclause 4.7.6, add the following text at the end of the subclause:

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). The following methods are defined:

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

Subclause 4.7.7, replace the subclause with the following text:

4.7.7 Colour attributes

In CGM colours are described by a colour model together with a specification of colour coordinates in the colour space of that model. Colour models define a colour coordinate system and a subspace, within which each describable colour is represented by a point. The CGM provides the following colour models: RGB (the default), CIELAB, CIELUV and CMYK. The selection of one of these models is made in the Metafile Descriptor.

RGB is an additive colour mixture system, i.e., the red (R), green (G) and blue (B) stimuli combine additively their radiant intensity together to form the complete range of colours. In case RGB data are non-linear in the radiant intensity, a look-up table may be used to transform it into (linear) RGB tristimulus values. RGB is used by a number of different types of devices, such as colour monitors, film writers, and colour input scanners.

Two CIE recommended uniform colour spaces, CIELAB and CIELUV, are allowed in the CGM. These colour spaces are non-linear transformations of the CIE 1931 XYZ tristimulus space, into the perceptual attributes of brightness and chroma. CIELAB and CIELUV closely approximate a uniform colour space over small distances, and provide an approximately uniform measure of perceived colour differences. CIELUV is commonly used for applications involving self-luminous displays where the additivity provided by its associated chromaticity diagram is important. CIELAB is more commonly used in surface colour applications and for the paints, plastics and textile industries.

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. In the CGM, the quantities C, M, Y, K represent the relative area occupied by a colorant at a particular point in order to produce a final image. In theory, the cyan, magenta and yellow colorants are complementary to the red, green and blue colorants. In practice, black colorant is added to increase the colour gamut (lower L^* values), i.e. higher density blacks are possible than with usual cyan, magenta and yellow colorants.

NOTE — The fact that ISO 8632 allows 3- and 4-dimensional colour spaces results in a 3-tuple or 4-tuple data type for direct colour specification, depending on the COLOUR MODEL element.

The meaning of each colour space allowed in the CGM (RGB, CIELAB, CIELUV and CMYK) is defined by the transformation necessary to convert a colour specification expressed in a reference colour space to or from the specification of the same colour in the CGM colour space (colour calibration). The reference colour space is the CIE 1931 XYZ space, which is defined in CIE Publication 15.2. Conceptually, colour values specified by one of the CGM colour spaces are interpreted by converting them into the reference colour space, and then converting them from the reference colour space to the device space of the interpreter.

NOTE — CIE Publication 15.2 defines the CIE 1931 XYZ space in terms of the CIE 1931 2° Standard Observer (CIE S002). The space is colorimetrically precise and covers all perceivable colours. It is based on properties of the human visual system, determined by extensive experiments in colour matching, rather than on the properties of any particular device. In 1964, the CIE defined a new 10° standard observer for matching colour fields from 4° to 10° in angular subtense. As smaller colour fields are expected in computer graphics, the 2° 1931 Standard Observer was selected.

NOTE — It is recognized that the general problem of appearance matching has not been completely solved. However, this standard uses the best available and internationally recognized approach, which is the CIE system of colorimetry

NOTE — The specification of a colour in the reference colour system can be made by reporting its XYZ tristimulus values and/or its x, y, z chromaticity coordinates, which are given by $x = X/T$, $y = Y/T$, where $T = X+Y+Z$. Plotting y versus x results in a horseshoe shape curve representing the chromaticities of the pure spectrum colours. If, for example, the colours available from an RGB system are transformed into chromaticity coordinates, the RGB system will be represented as a triangle contained within the horseshoe.

The reference colour space is normalized such that the Y tristimulus value is 1 for the reference white to allow for simplicity of conversions from colorimetric values to other colour spaces.

NOTE — This differs from the CIE recommendation of normalizing Y of the perfect reflecting (or transmitting) diffuser (reference white) to exactly 100.

The colour calibration depends on the colour space specification. The calibration data for all colour spaces contain a reference white value to allow for how colours are percieved in relation to the viewing environment. This is the set of CIEXYZ values of the reference white (X_n, Y_n, Z_n).

The reference white value is the only calibration data applicable to the CIELAB and CIELUV colour spaces.

Additionally, the calibration data for the RGB colour space consists of a 3x3 matrix, specifying the position of the RGB primaries in CIEXYZ space.

The calibration data for the CMYK colour space consist of CMYK grid locations and corresponding CIEXYZ values. No calibration data are available for standard ink set.

For details of conversion between each CGM colour space and the CIE reference colour space, see Annex J.

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 specified 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.

Concepts

For direct colour specification in RGB and CMYK colour spaces, normalized weights for the colour components 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 space, 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.

For direct colour specification in CIELAB and CIELUV colour spaces, each component of a 3-tuple is represented through a scale and offset factor.

NOTE — For example, a defined colour component value labelled x is represented by s*x + o, s and o being the scale and offset.

No range is specified as in the RGB and CMYK colour spaces. This is due to the fact that the visible colours do not lie within the unit cube as for RGB and CMYK colour spaces.

NOTE — In case of CIELUV colour space, for example, in the abstract, the visible colours lie within a cone with apex at the origin. The locus of the curve in u, v enclosing all colours at a given lightness L depends on the chosen reference white.

There is a Metafile Descriptor element, COLOUR VALUE EXTENT, which allows metafile generators to specify the minimum and maximum metafile colour values for RGB and CMYK colour spaces; these correspond with the abstract (0,0,0) and (1,1,1). For CIELAB and CIELUV colour spaces, this element specifies the scale and offset parameters.

Page 38

Subclause 4.7.8, add the following text to the end of the description of 'hatch':

Hatch styles may be user defined in the CGM. An arbitrarily complex arrangement of hatch lines and inter-line spaces may be defined, and direction vectors for either single hatch or cross hatch may be specified. The colour and line type of the lines in the hatch may be defined as a part of the hatch style definition using the HATCH STYLE DEFINITION element. Hatch styles are associated with an index when defined, and invoked as a filling interior with the normal HATCH INDEX element when the interior style is ***hatch***. Changing the definition of an already defined hatch index shall have no retroactive or dynamic effects.

Page 38

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 38

Subclause 4.7.8, add the following text to the end of the description of 'pattern':

...or the geometric pattern table.

Page 38

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.

geometric pattern: fill the interior using the geometric pattern associated with the pattern index currently selected (either via PATTERN INDEX or FILL BUNDLE INDEX, depending on the corresponding PATTERN INDEX ASF). The GEOMETRIC PATTERN element associates the index with a segment which is used to fill the interior.

Page 38

Subclause 4.7.8, add new subclause:

4.7.8.1 Interpolated Interiors

The metafile provides a general purpose element for defining several different styles of interpolated interiors. A solid but continuously graded colour interior is defined for filled-area primitives. Conceptually a graded-colour plane of infinite extent is defined and this is "extruded" through the interior of filled area primitives to define the appearance of the interior.

A number of styles are defined. For each style the parameterization of the element defines a "reference geometry" — two parallel lines, an ellipse, or a triangle — and then defines colour interpolation points whose positions are defined relative to the reference geometry. Colours in the parameter list of the element are assigned to these latter points. For some of the styles (parallel and elliptical) multiple parallel or concentric bands (stages) of independently interpolated colour may be defined.

The following styles are defined:

parallel: This is a multi-stage style. Each stage is a semi-infinite band bounded by two parallel lines. A reference colour is defined on each parallel line. Colour is constant along any parallel line within the band and is equal to the linear interpolant of the reference colours. Outside of the outermost defined bands (in the two semi-infinite half planes) the colour is constant and equal to the corresponding outer-most reference colours.

elliptical: This is a multi-stage style. Each stage is an elliptical annulus with a reference colour defined on each of the inner and outer bounding ellipses. Colours are constant along ellipses which are concentric to the reference ellipse. The colour at any ellipse within a band is constant and equal to the linear interpolant of the reference colours on the two bounding ellipses. Outside of the outer-most defined ellipse the colour is constant and equal to the corresponding outer-most reference colour.

triangular: Colours are associated with three points defining a triangle. The unique bi-linear interpolated colour is defined at each point in the interior of the triangle. Outside of the triangle the colour is defined to be constant on rays from the centre of the triangle to infinity and equal to the interpolated colour value on the boundary of the triangle.

The parameterization of the element defining interpolated interiors is consistent with application of the rules of INTERIOR STYLE SPECIFICATION MODE regarding transformation of interiors. The reference geometry and style defined by the element transform or do not transform according to the value of the mode.

Colour interpolation is performed in the colour space specified by the COLOUR MODEL element.

Page 38

After subclause 4.7.9 (in Am.1), add:

4.7.10 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 through the end of a straight line segment and into 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. If two segments which are adjacent in the definition are not physically contiguous, i.e., the end point of the first one does not coincide with the first point of the next one, then the path includes the straight line segment joining these two points.

4.7.11 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). If two segments which are adjacent in the definition are not physically contiguous, i.e., the end point of the first one does not coincide with the first point of the next one, then the path includes the straight line segment joining these two points.

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 ghyph's reference point aligns with the junction of two line elements of the compound text path, it is implementation dependent whether the final direction of the first segment or the initial direction of the second segment will be used. Positioning of subsequent glyphs is accomplished so that the arc length between the glyph position points is the same as the distance between glyph position points when they are laid out on a straight line. When the path ends before all glyphs have been placed, the path for the excess text is the straight line described by the tangent at the end of the compound text path.

NOTE — Using paths with sharp curvature is likely to cause overlapping tops or bottoms of characters unless care is taken in adjusting the text attributes.

4.7.12 Picture Composition

The picture composition elements are:

BEGIN PROTECTION REGION
END PROTECTION REGION
PROTECTION REGION INDICATOR

In addition, CLIP RECTANGLE and CLIP INDICATOR may be used for confining graphical output to simple rectangular areas of the display surface.

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.

Protection regions are identified by an index. Multiple regions may be active simultaneously. Protection regions are constructed by the same primitive elements as closed figures. The interior of a given protection 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 PROTECTION 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.

Protection regions behave as do clip rectangles with respect to transformations — they transform by the copy transformation which is associated with COPY SEGMENT.

If separate protection regions are simultaneously active, then the aggregate protection 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.

A protection region is associated with an index at definition time. If a region is defined with an index that is already in use then the old definition is deleted. The associated protection region indicator remains as specified.

4.7.13 Line Attributes

In addition to the line attributes which may be bundled there are a number of individual line attributes.

a) LINE CAP specifies the appearance of the endpoints of line elements as well as the endpoints of individual dashes when dashed lines are rendered. The following cap styles are supported:

unspecified:	As in ISO 8632 version 1, any implementation dependent treatment is acceptable.
butt:	The line is squared off at the endpoint, there is no projection beyond the endpoint.
round:	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.
projecting square:	The line is squared off at a distance equal to half the line width beyond the endpoint.
triangle:	A cap is added to the line which is an equilateral triangle whose side equals the line width.

These styles may be applied to the open endpoints of line elements — those endpoint which demark the beginning or end of the entire line primitive — or to interior endpoints, which correspond to the endpoints of individual dashes when a non-solid line type is in effect. The interior caps must either match the caps on the open endpoints or be implementation dependent. Figure 11j illustrates the styles of LINE CAP.

b) LINE JOIN specifies the appearance of the interior corners of line elements. Interior corner correspond either to the interior vertices of polyline elements or to the junctions between distinct elements comprising a compound line element. The following styles are supported:

unspecified:	As in ISO 8632 version 1, any implementation dependent treatment is acceptable.
mitre:	The outer edges of the two adjoining line segments are extended until they meet at a point.
round:	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.
bevel:	The adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

The rendering of the line join is affected by the MITRE LIMIT Control Element. Figure 11j illustrates the defined styles of LINE JOIN.

Both line caps and line joins behave as does line width with respect to transformation. If the value of LINE WIDTH SPECIFICATION MODE is ***absolute*** then conceptually the line cap and join are applied to the line in VDC space before any transformations are applied and they are subject to all transformations associated with the line. Otherwise they are applied to the line in device space, conceptually after all associated transformations have been applied, and are immune to all transformations.

c) LINE TYPE CONTINUATION provides control of the behaviour of non-solid line types at interior vertices and junctions of line elements. The following behaviours may be selected:

unspecified: As in ISO 8632 version 1, any implementation dependent continuation is acceptable;

continue: The style is continued without interruption across vertices;

restart: The style is restarted at each vertex;

adaptive continue: The style is continued, but each vertex must be "inked" including vertices at the ends of the line primitive which might otherwise not be drawn because of a non-solid line type.

d) LINE TYPE INITIAL OFFSET allows control of how much of the first cycle of a non-solid line type is skipped before drawing commences for a line primitive. It is specifies as a fraction of one full cycle.

e) LINE AND EDGE TYPE DEFINITION allows the precise definition of the solid/gap sequences which comprise a line or edge type. A definition is associated with an index by the element. If an already defined index is redefined, there are no retroactive or dynamic effects — the redefinition only affects subsequently occurring line or edge elements.

4.7.14 Edge Attributes

In addition to the edge attributes which may be bundled there are a number of individual edge attributes.

a) EDGE CAP specifies the appearance of the endpoints of edges, such as might occur due to the turning off and on of edge visibility in POLYGON SET elements or CLOSED FIGURE elements. The supported styles and their definitions are as for the LINE CAP element, with the endpoints of visible edge segments corresponding to the "open endpoints" of the line elements definition.

b) EDGE JOIN specifies the appearance of edges at the vertices of filled-area elements or at junctions between distinct elements in compound filled area elements. The supported styles and their definitions are as for the LINE JOIN element. The rendering of the edge join is affected by the MITRE LIMIT Control Element.

   Both edge caps and edge joins behave as does edge width with respect to transformation. The behaviour is determined by the value of EDGE WIDTH SPECIFICATION MODE, and is as described for LINE CAP and LINE JOIN.

c) EDGE TYPE CONTINUATION provides control of the behaviour of non-solid edge types at interior vertices and junctions of the edges of filled-area elements. The supported styles and their definitions are as for the LINE TYPE CONTINUATION element.

d) EDGE TYPE INITIAL OFFSET allows control of how much of the first cycle of a non-solid line type is skipped before drawing commences for a line primitive. It is specifies as a fraction of one full cycle.

Page 40 (Am.1, page 18)

In 4.12.1, first paragraph, add:

d) visibility

Page 40 (Am.1, page 18)

In 4.12.1, last paragraph, add to the list:

SEGMENT VISIBILITY

Page 40 (Am.1, page 18)

In 4.12.2, 4th sentence, add at end:

...if segment visibility is *on*.

Page 40 (Am.1, page 20)

Insert new section 4.12.4.6:

4.12.4.6 Segment visibility. The segment visibility attribute specifies whether a segment is visible or not when it is defined. This attribute only affects Local Segments. Global Segments are never visible when defined.

Page 40 (Am.1, page 20)

Insert in 4.12.5 after first sentence:

The segment attributes are not inserted into the picture.

Page 40 (Am.1, page 21)

In 4.12.5 add to the LINE ATTRIBUTES of table 3a INHERITANCE FILTER:

LINE CAP
LINE JOIN
LINE TYPE CONTINUATION
LINE TYPE INITIAL OFFSET

In 4.12.5 add to the TEXT PRESENTATION AND PLACEMENT ATTRIBUTES of table 3a INHERITANCE FILTER:

TEXT SCORE TYPE
RESTRICTED TEXT TYPE

In 4.12.5 add to the TEXT PRESENTATION AND ORIENTATION ATTRIBUTES of table 3a INHERITANCE FILTER:

GENERALIZED TEXT PATH MODE

In 4.12.5 add to the FILL ATTRIBUTES of table 3a INHERITANCE FILTER:

INTERPOLATED INTERIOR DEFINITION

In 4.12.5 add to the EDGE ATTRIBUTES of table 3a INHERITANCE FILTER:

EDGE CAP
EDGE JOIN
EDGE TYPE CONTINUATION
EDGE TYPE INITIAL OFFSET

In 4.12.5 add after PICK IDENTIFIER of table 3a INHERITANCE FILTER:

SYMBOL ATTRIBUTES	SYMBOL LIBRARY INDEX
	SYMBOL COLOUR
	SYMBOL SIZE
	SYMBOL ORIENTATION

In 4.12.5 add to OUTPUT CONTROL of table 3a INHERITANCE FILTER:

MITRE LIMIT

Page 40 (Am.1, page 23)

In 4.12.5 add sentence at the top of the page:

The description of clipping in this subclause also applies to the protection region used for clipping and shielding of areas.

Page 41

Replace the Amendment 1 State Table, Table 3c, which follows the State Diagram, with:

	CGM Higher States						
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 COMPOUND PATH				X		X	X
+END COMPOUND PATH							
+BEGIN PROTECTION REGION				X		X	X
+END PROTECTION REGION							
+BEGIN TILE ARRAY						X	
+END TILE 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 (continued)						
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		
INTERIOR STYLE SPECIFICATION MODE			X	X	X	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
+PROTECTION REGION INDICATOR			X	X		X	X
+MITRE LIMIT			X	X		X	X
+GENERALIZED TEXT PATH MODE			X	X		X	X

	CGM Higher States (continued)						
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							
+HYPERBOLIC ARC				X		X	X
+PARABOLIC ARC				X		X	X
+NON-UNIFORM B-SPLINE				X		X	X
+NON-UNIFORM RATIONAL B-SPLINE				X		X	X
+POLYBEZIER				X		X	X
+SYMBOL				X		X	X
+BITONAL TILE						X	
+TILE						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

	CGM Higher States (continued)						
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	
COLOUR TABLE			X		X	X	
ASPECT SOURCE FLAGS			X	X		X	X
PICK IDENTIFIER			X	X		X	X
+LINE & EDGE TYPE DEFINITION			X	X		X	X
+LINE CAP			X	X		X	X
+LINE JOIN			X	X		X	X
+LINE TYPE CONTININUATION			X	X		X	X
+LINE TYPE INITIAL OFFSET			X	X		X	X
+TEXT SCORE TYPE			X	X		X	X
+RESTRICTED TEXT TYPE			X	X		X	X
+HATCH STYLE DEFINITION			X	X		X	X
+GEOMETRIC PATTERN				X		X	X
+INTERPOLATED INTERIOR DEFINITION			X	X		X	X
+EDGE CAP			X	X		X	X
+EDGE JOIN			X	X		X	X
+EDGE TYPE CONTININUATION			X	X		X	X
+EDGE TYPE INITIAL OFFSET			X	X		X	X
+SYMBOL LIBRARY INDEX			X	X		X	X
+SYMBOL COLOUR			X	X		X	X
+SYMBOL SIZE			X	X		X	X
+SYMBOL ORIENTATION			X	X		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
SEGMENT VISIBILITY				X			X

	CGM Lower States				
CGM Element	FOS	TOS	PHS	PRS	TAS
BEGIN METAFILE					
BEGIN PICTURE					
BEGIN PICTURE BODY					
END PICTURE					
BEGIN SEGMENT					
END SEGMENT					
BEGIN FIGURE					
END FIGURE	X				
END METAFILE					
+BEGIN COMPOUND PATH					
+END COMPOUND PATH			X		
+BEGIN PROTECTION REGION					
+END PROTECTION REGION				X	
+BEGIN TILE ARRAY					
+END TILE 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 (continued)				
CGM Element	FOS	TOS	PHS	PRS	TAS
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					
INTERIOR STYLE SPECIFICATION MODE					
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					
+PROTECTION REGION INDICATOR					
+MITRE LIMIT					
+GENERALIZED TEXT PATH MODE					

CGM Element	CGM Lower States (continued)				
	FOS	TOS	PHS	PRS	TAS
POLYLINE	X		X	X	
DISJOINT POLYLINE	X		X	X	
POLYMARKER					
TEXT					
RESTRICTED TEXT					
APPEND TEXT		X			
POLYGON	X			X	
POLYGON SET	X			X	
CELL ARRAY					
GDP	X		X	X	
RECTANGLE	X			X	
CIRCLE	X			X	
CIRCULAR ARC 3 POINT	X		X	X	
CIRC.ARC 3 POINT CLOSE	X			X	
CIRCULAR ARC CENTRE	X		X	X	
CIRCULAR ARC CENTRE CLOSE	X			X	
ELLIPSE	X			X	
ELLIPTICAL ARC	X		X	X	
ELLIPTICAL ARC CLOSE	X			X	
CIRCULAR ARC CENTRE REVERSED	X		X	X	
CONNECTING EDGE	X				
+HYPERBOLIC ARC	X		X	X	
+PARABOLIC ARC	X		X	X	
+NON-UNIFORM B-SPLINE	X		X	X	
+NON-UNIFORM RATIONAL B-SPLINE	X		X	X	
+POLYBEZIER	X		X	X	
+SYMBOL					
+BITONAL TILE					X
+TILE					X
LINE BUNDLE INDEX					
LINE TYPE					
LINE WIDTH					
LINE COLOUR					
MARKER BUNDLE INDEX					
MARKER TYPE					
MARKER SIZE					
MARKER COLOUR					
TEXT BUNDLE INDEX		X			
TEXT FONT INDEX		X			
TEXT PRECISION		X			
CHARACTER EXPANSION FACTOR		X			
CHARACTER SPACING		X			
TEXT COLOUR		X			
CHARACTER HEIGHT		X			
CHARACTER ORIENTATION					
TEXT PATH					
TEXT ALIGNMENT					
CHARACTER SET INDEX		X			
ALTERNATE CHARACTER SET INDEX		X			

	CGM Lower States (continued)				
CGM Element	FOS	TOS	PHS	PRS	TAS
FILL BUNDLE INDEX					
INTERIOR STYLE					
FILL COLOUR					
HATCH INDEX					
PATTERN INDEX					
EDGE BUNDLE INDEX	X				
EDGE TYPE	X				
EDGE WIDTH	X				
EDGE COLOUR	X				
EDGE VISIBILITY	X				
FILL REFERENCE POINT					
PATTERN TABLE					
COLOUR TABLE					
ASPECT SOURCE FLAGS	X				
PICK IDENTIFIER					
+LINE & EDGE TYPE DEFINITION					
+LINE CAP					
+LINE JOIN					
+LINE TYPE CONTININUATION					
+LINE TYPE INITIAL OFFSET					
+TEXT SCORE TYPE					
+RESTRICTED TEXT TYPE					
+HATCH STYLE DEFINITION					
+GEOMETRIC PATTERN					
+INTERPOLATED INTERIOR DEFINITION					
+EDGE CAP	X				
+EDGE JOIN	X				
+EDGE TYPE CONTININUATION	X				
+EDGE TYPE INITIAL OFFSET	X				
+SYMBOL LIBRARY INDEX					
+SYMBOL COLOUR					
+SYMBOL SIZE					
+SYMBOL ORIENTATION					
ESCAPE	X	X	X	X	X
MESSAGE	X		X	X	
APPLICATION DATA	X		X	X	X
COPY SEGMENT					
INHERITANCE FILTER					
CLIP INHERITANCE					
SEGMENT TRANSFORMATION					
SEGMENT HIGHLIGHTING					
SEGMENT DISPLAY PRIORITY					
SEGMENT PICK PRIORITY					
SEGMENT VISIBILITY					

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
PHS Path State
PRS Protection Region State
TAS Tile Array State

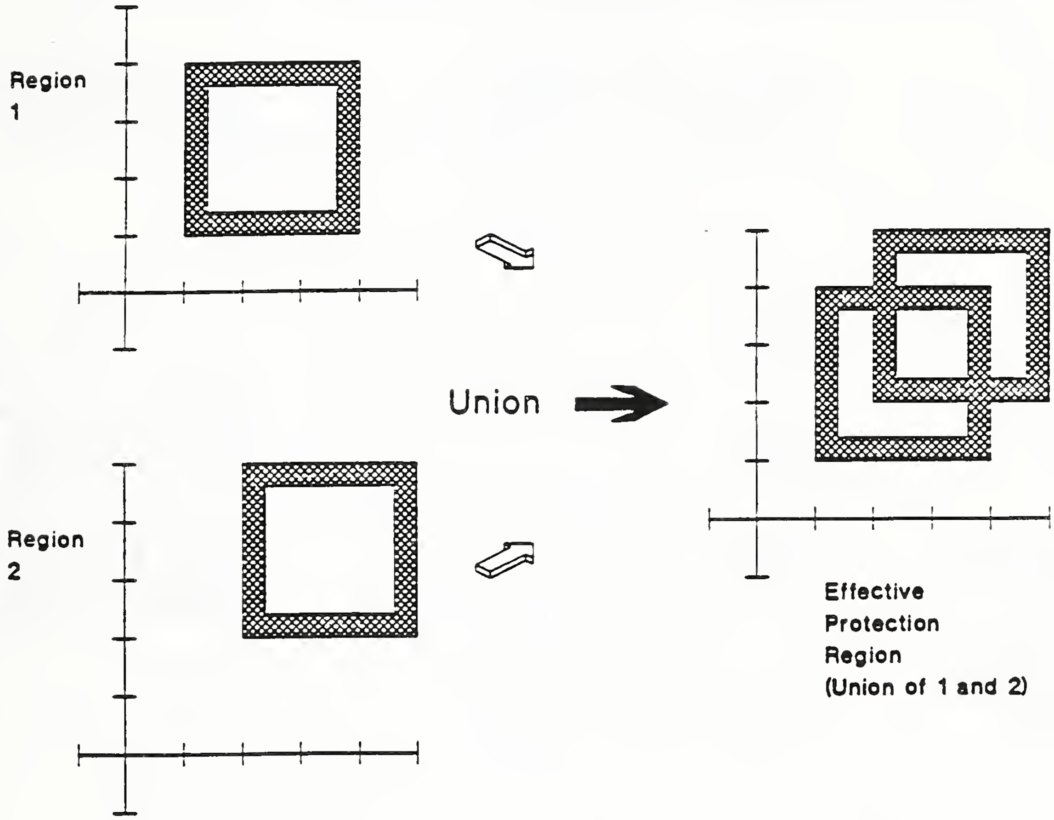

Figure 1b — Combining protection regions.

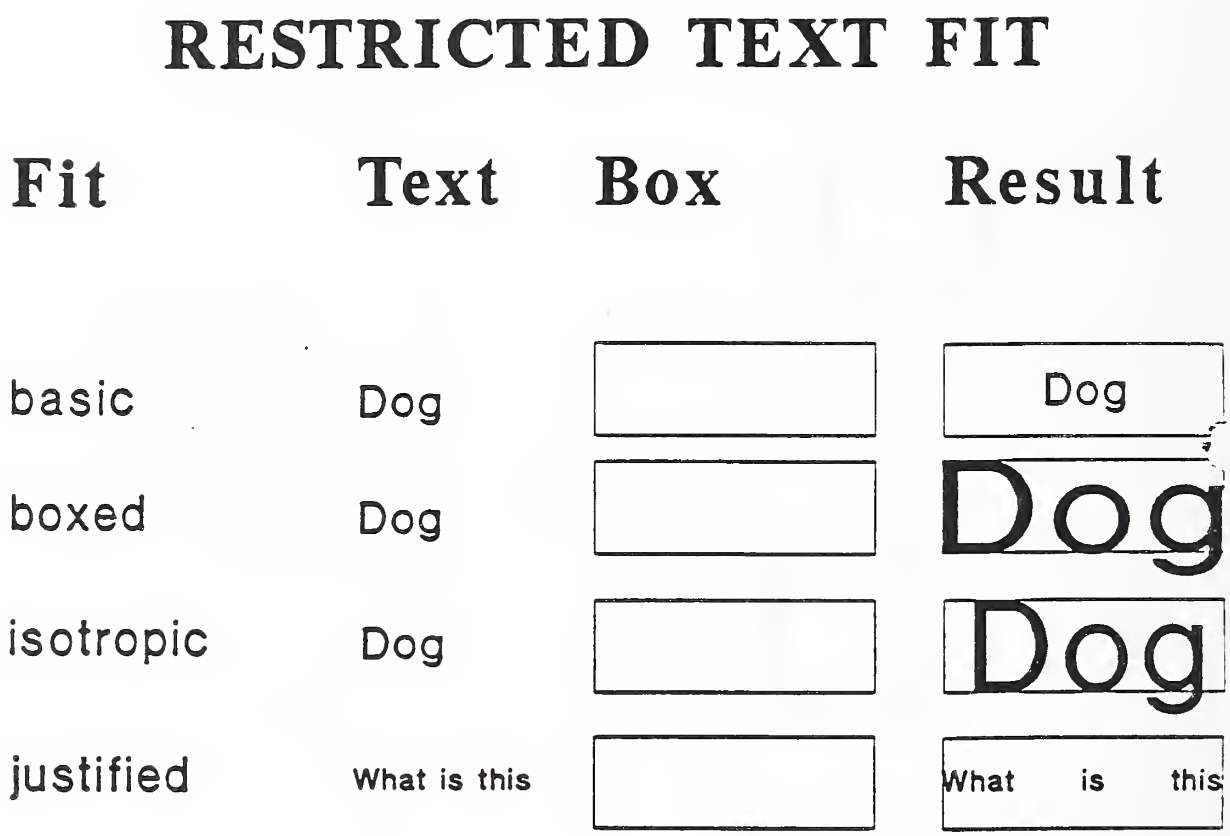

Figure 11a — Examples of RESTRICTED TEXT TYPE.

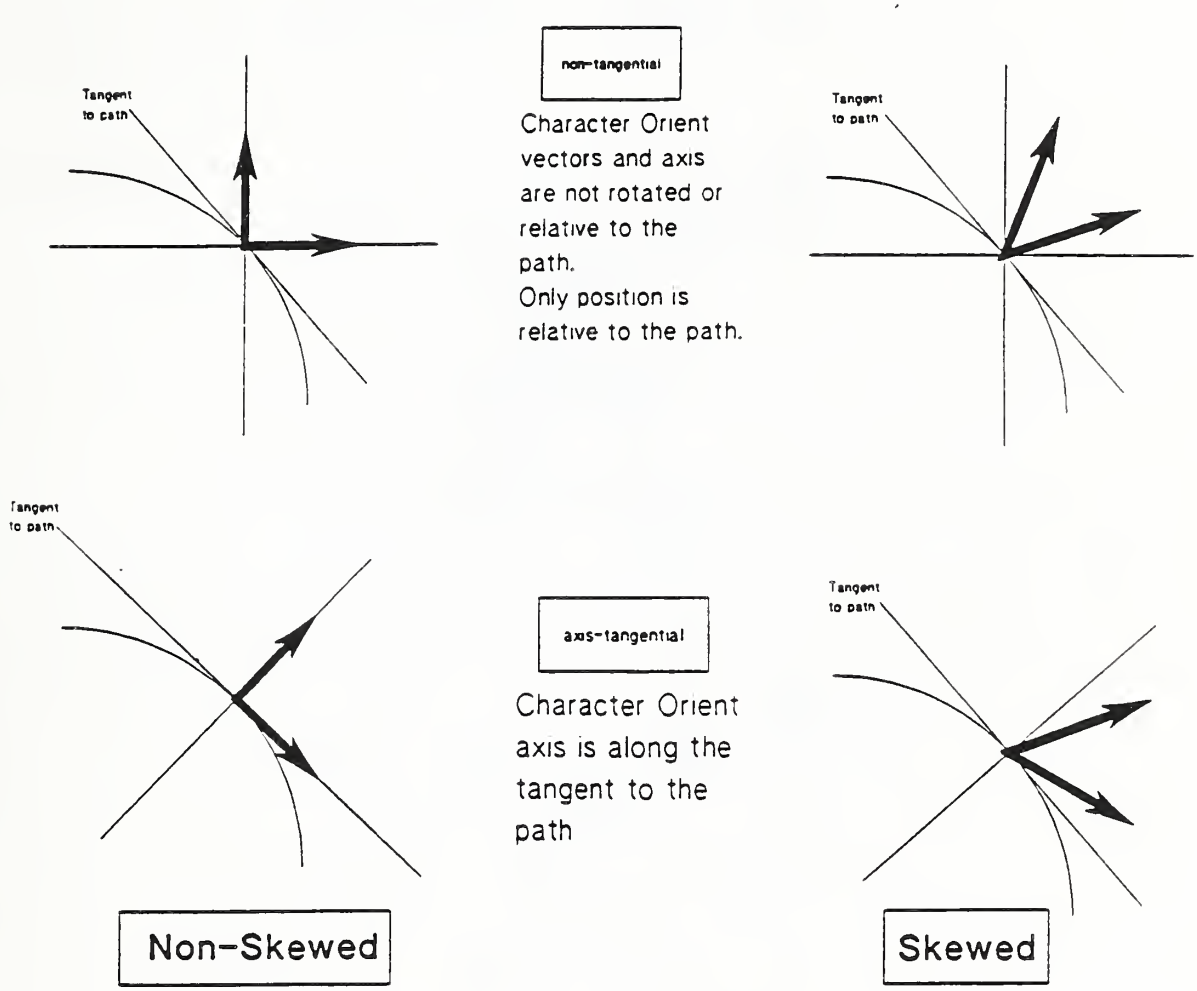

Figure 11b — Illustration of the modes of GENERALIZED TEXT PATH MODE.

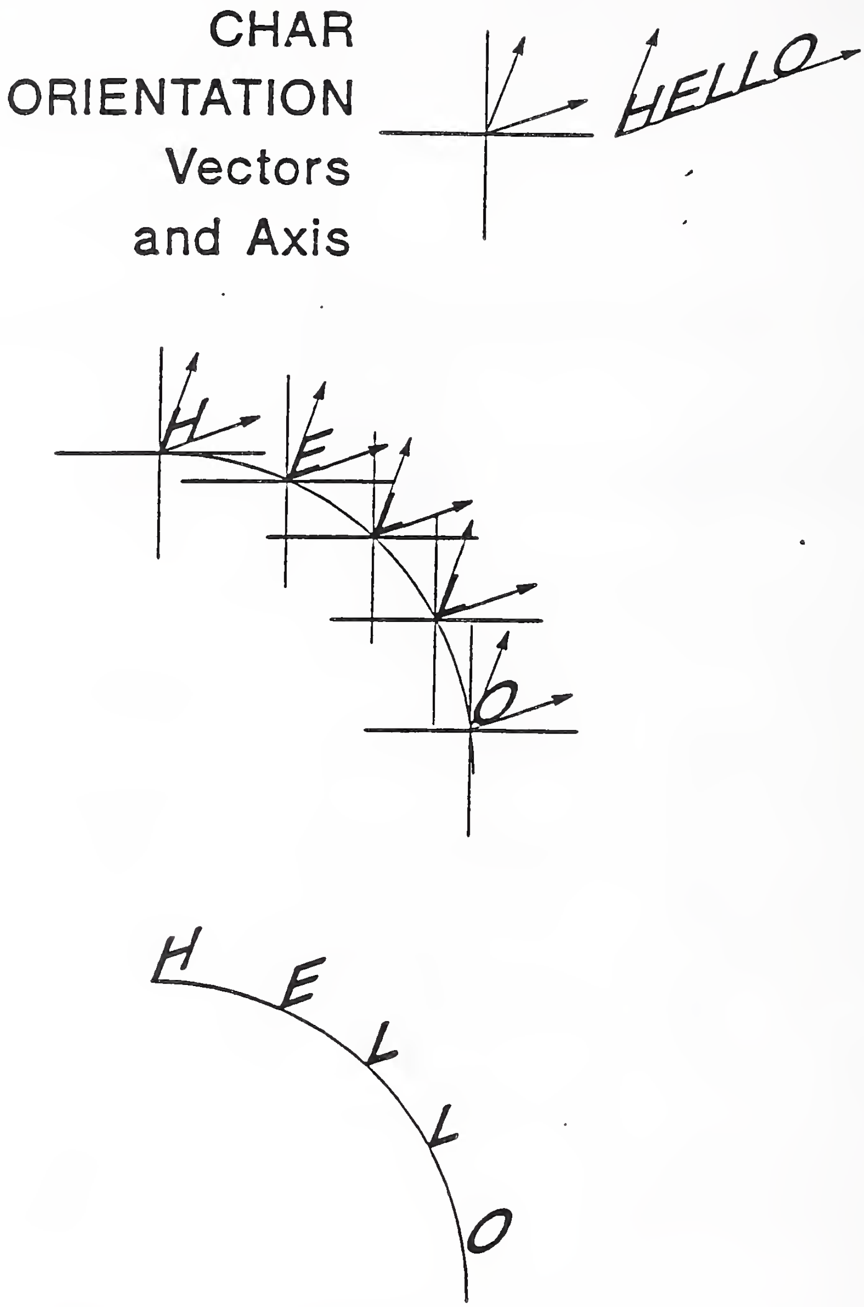

Figure 11d — Examples of GENERALIZED TEXT PATH MODE, non-tangential.

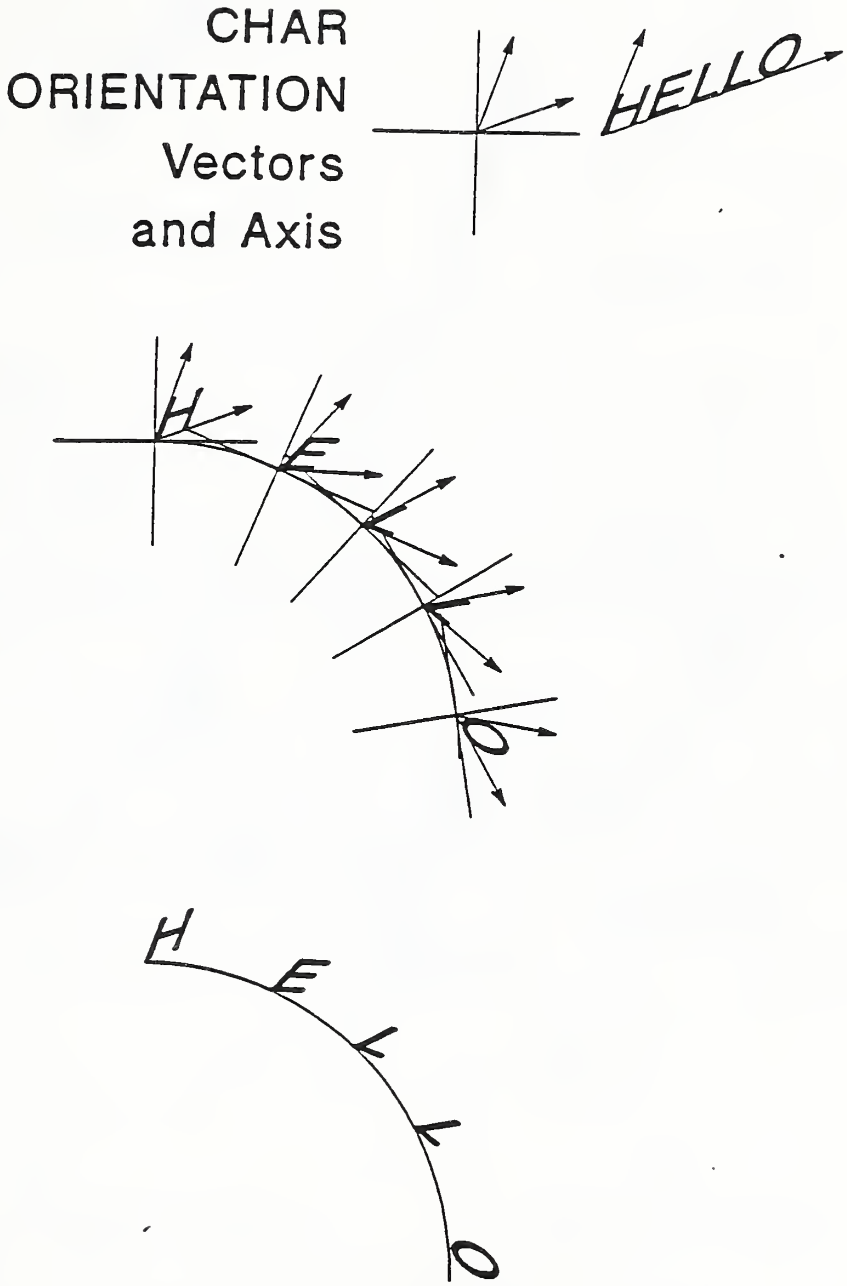

Figure 11e — Examples of GENERALIZED TEXT PATH MODE, axis-tangential.

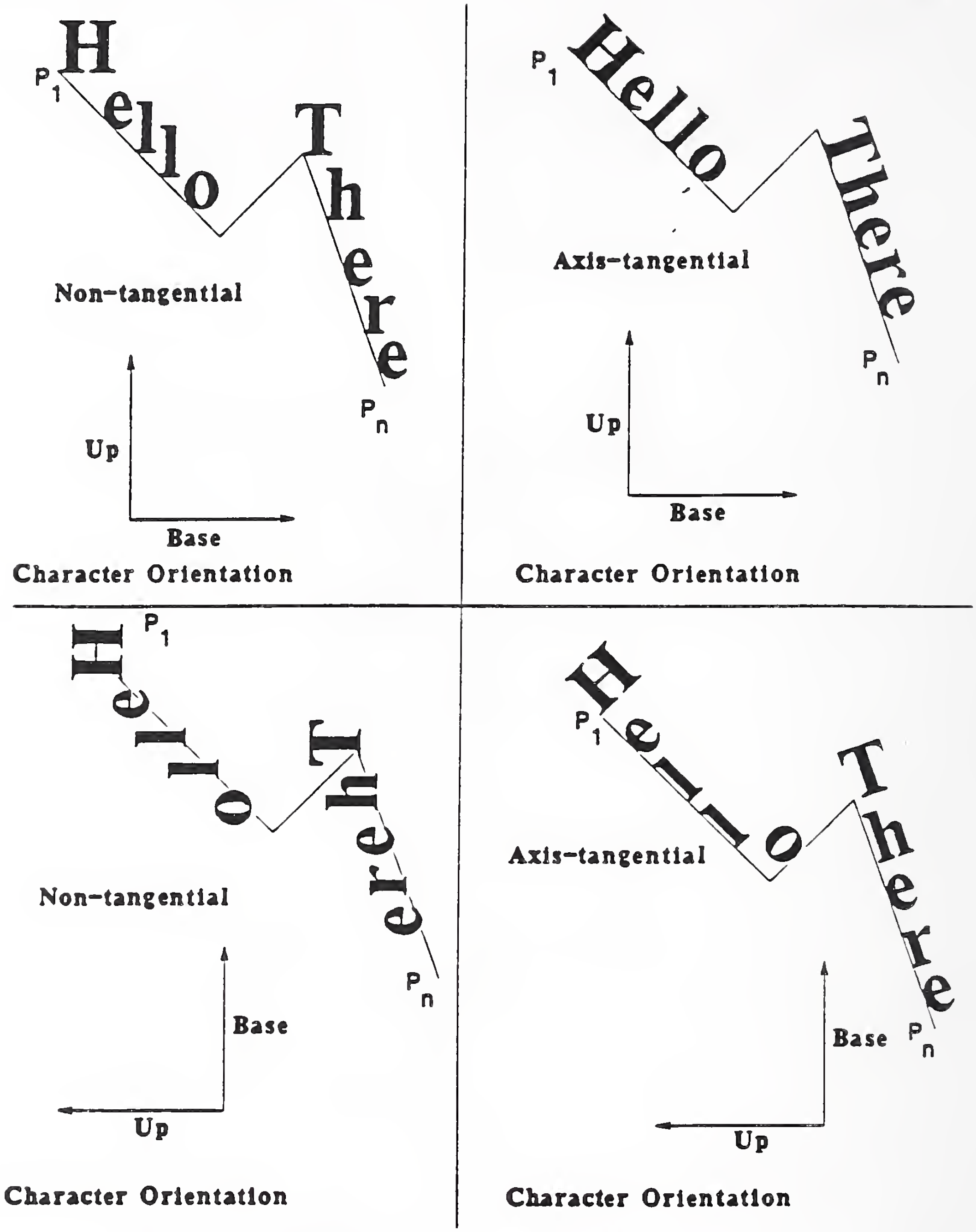

Figure 11f — GENERALIZED TEXT PATH MODE and unsmooth paths.

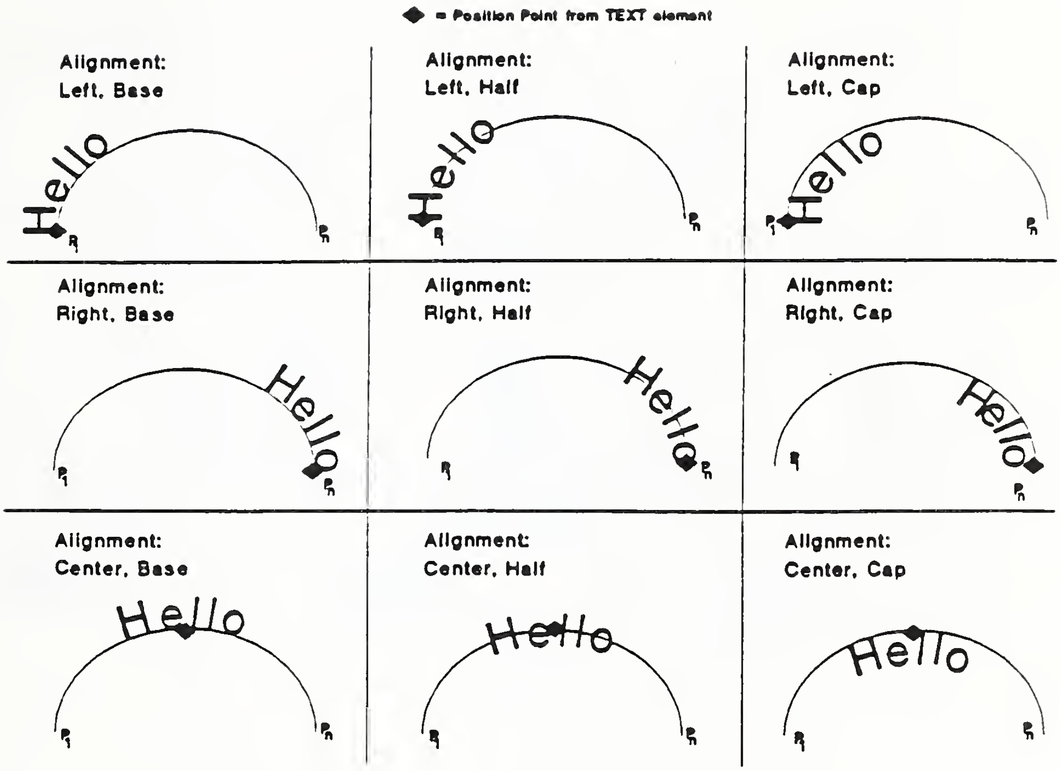

Figure 11g — Effect of TEXT ALIGNMENT and path text.

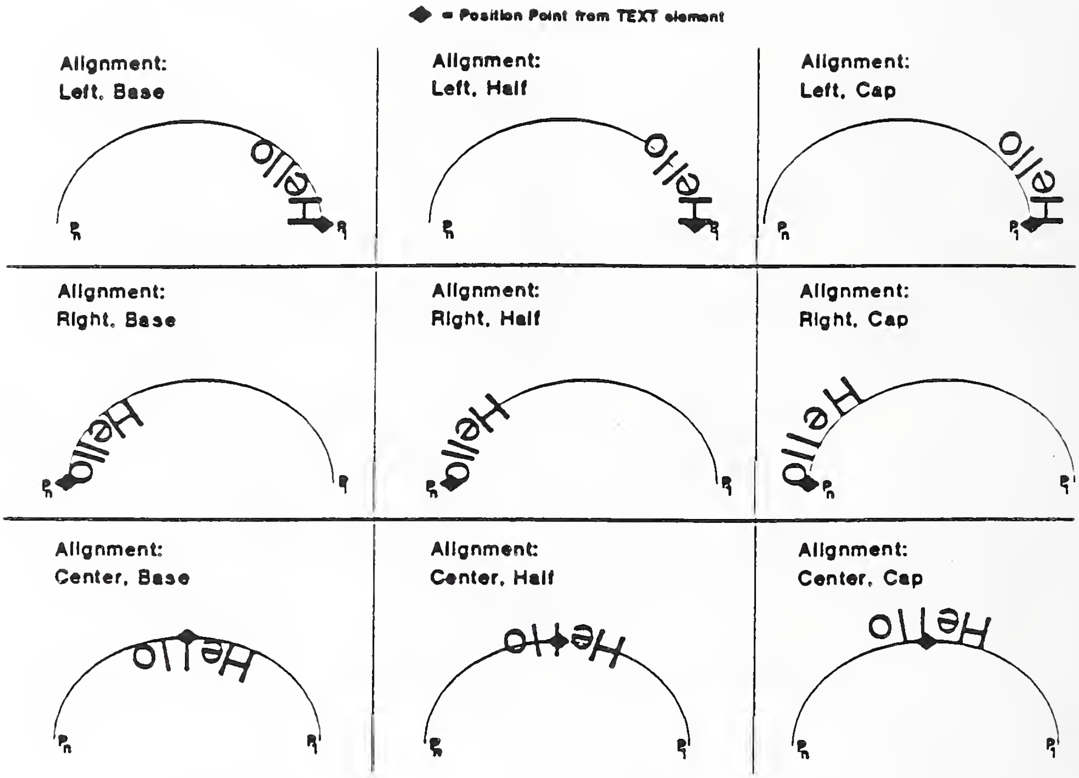

Figure 11h — Effect of path direction and TEXT ALIGNMENT with path text.

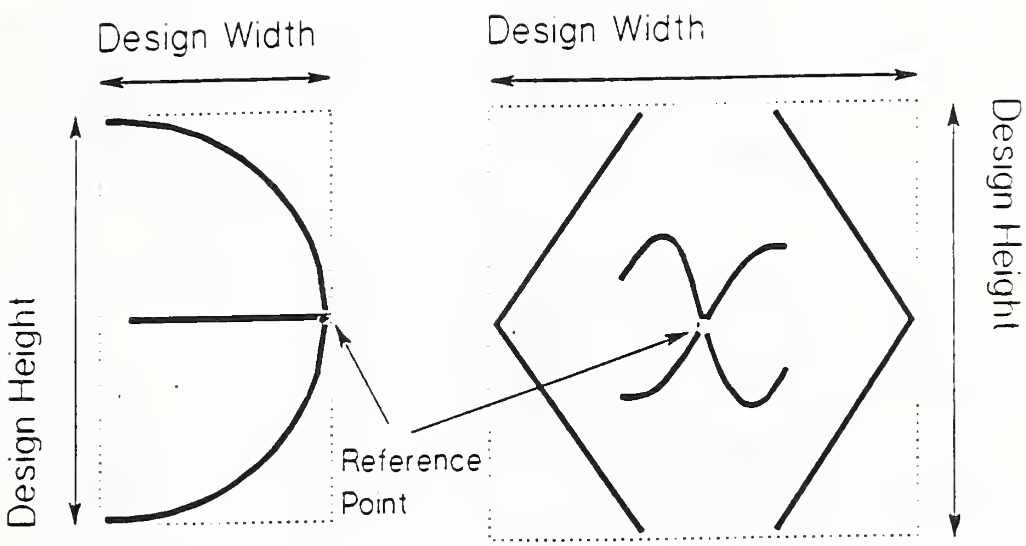

Figure 11i — Symbol coordinate system.

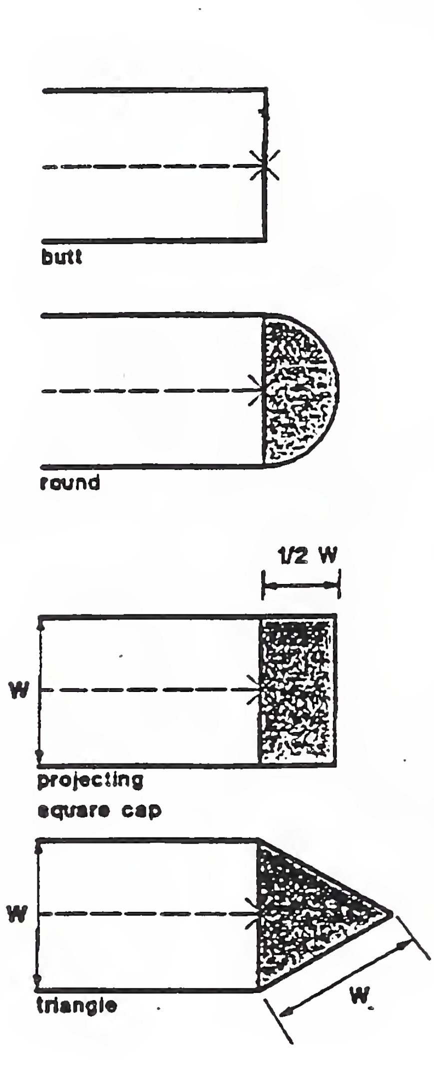

Figure 11j — Examples of LINE CAP and LINE JOIN.

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 CCO values (as determined by COLOUR MODEL) for colour definition within one of the supported colour models.

Page 42

Subclause 5.1, table of data type abbreviations, add:

CCO	Colour component	one component of a colour direct value.
OC	Octet	an unsigned integer in the range 0..255, represented in each of the encodings with a precision equivalent to 8 binary bits.
I32	32-bit Integer	an unsigned integer in the range $0..(2^{32}-1)$, represented in each of the encodings with a precision equivalent to 32 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:

5.2.10 BEGIN COMPOUND PATH

Parameters:

path type (IX)

Description:

The path type of the defined compound path is selected. The defined values are:

1: text path;
2: compound line.

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:
4.2
4.7.6
4.7.10
4.7.11

5.2.11 END COMPOUND PATH

Parameters:

None

Description:

END COMPOUND PATH delimits the end of a compound line or text path definition.

References:
4.2
4.7.6
4.7.10
4.7.11

5.2.12 BEGIN PROTECTION REGION

Parameters:

region index (IX)

Description:

Line and fill primitives which are present between the BEGIN PROTECTION REGION and END PROTECTION REGION are used to construct a protection region. The region is used either for clipping or for shielding, as specified by the PROTECTION REGION INDICATOR element. The defined region is associated with the region index parameter, by which it may subsequently be referenced by the PROTECTION REGION INDICATOR element.

References:
4.2

Elements

4.5.4
4.7.12

5.2.12 END PROTECTION REGION

Parameters:

None

Description:

END PROTECTION REGION delimits the end of a protection region definition.

References:
4.2
4.5.4
4.7.12

5.2.14 BEGIN TILE ARRAY

Parameters:

position (P)
cell path direction (one of 0, 90, 180, 270) (E)
line progression direction (one of 90, 270) (E)
number of tiles in path direction (I)
number of tiles in line direction (I)
number of cells/tile in path direction (I)
number of cells/tile in line direction (I)
cell size in path direction (R)
cell size in line direction (R)
image offset in path direction (I)
image offset in line direction (I)
image number of cells in path direction (I)
image number of cells in line direction (I)

Description:

A tile array is defined as follows:

The *position* is used to place the tile array. The first cell of the first tile is placed at the specified point.

The *cell path direction* parameter defines the direction of progression of successive cells along a line relative to the VDC x-axis. The *line progression direction* parameter defines the direction of progression of successive cell lines and is expressed as a direction relative to the cell path direction.

The *cell size in path direction* and *cell size in line direction* are defined in units of number of cells per VDC unit.

The *cell size in path direction* and *number cells/tile path direction* parameters together define the length and granularity for each line in the tile, hence the tile size in the cell path direction. The *line size in path direction* and *number cells/tile in line direction* parameters together implicitly define the tile size in the line progression direction.

When laid out all of the tiles in the tile array define a rectangular sub-region of VDC space — a "tiling space". The actual graphical image may not (in fact in large tiled images will not) occupy the full rectangle. The image offset and image number of cells parameters specify the rectangle within the tiling space which is actually occupied by the image. Cells whose position is outside of this rectangle are not even encoded. The number of encoded cells in a tile is thus dependent upon the tile location and the image offset and size parameters.

References:
4.2
4.6.5.1
D.4.6

5.2.15 END TILE ARRAY

Parameters:

None

Description:

This element terminates the definition of tile array which was commenced by BEGIN TILE ARRAY.

References:
4.2
4.6.5.1
D.4.6

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 description as follows:

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

Page 49

Subclause 5.3.10, COLOUR VALUE EXTENT replace as follows:

Elements

5.3.10 COLOUR VALUE EXTENT

Parameters:

colour value mapping specifier

if the colour model indicator is RGB or CMYK,
minimum colour value (CD)
maximum colour value (CD)

if the colour model indicator is CIELAB or CIELUV,
3 pairs of colour scale and colour offset (6R)

Description:

For colour model indicator RGB and CMYK, 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 values contained in the metafile. The 'minimum colour value' and 'maximum colour value' 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 RGB or CMYK selected for use in the metafile.

For colour model indicator CIELAB and CIELUV, the parameters represent the scale and offset that relate each component of a direct colour value to the colour value of the corresponding colour space. The three pairs apply to the first, second and third component of the colour direct value.

NOTE — For example, if CD(i), i=1,2,3, denotes the ith component of the colour direct value, then the corresponding colour value in CIELAB or CIELUV colour space is obtained through scale(i)*CD(i) + offset(i), where scale(i) and offset(i), i=1,2,3, denote the colour scale and colour offset for the ith colour component.

References:
4.7.7

Page 55

Subclause 5.3, add the following Metafile Descriptor Elements:

5.3.19 COLOUR MODEL

Parameters:

colour model indicator (IX)

Description:

The colour model of the metafile is selected. Standardized values include:

1: RGB
2: CIELAB
3: CIELUV
4: CMYK

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, representation setting elements, colour lists, 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

5.3.20 COLOUR CALIBRATION

Parameters:

reference white value (X_s, Y_s, Z_s) (3R)
calibration data

if the colour model indicator is RGB,
matrix (3x3)(R)
n (I)
array of pairs (R,R') of red components (n(2CCO))
array of pairs (G,G') of green components (n(2CCO))
array of pairs (B,B') of blue components (n(2CCO))

if the colour model indicator is CMYK,
n (I)
array of grid locations (CMYK) (nCD)
array of grid values (XYZ) (n(3R))

Description:

Colour calibration supplies the information which defines the transformation from the colour space selected by the COLOUR MODEL element to the CIEXYZ reference colour space.

The reference white value specifies the CIEXYZ values (X_s, Y_s, Z_s) of the reference white. The reference white value is the only colour calibration applicable to the CIELAB and CIELUV colour space for conversion to the CIEXYZ reference colour space.

When colour model indicator is RGB, the calibration data parameter specifies the values used to position the Red, Green, and Blue colour components in the CIEXYZ reference colour space. The matrix consists of the tristimulus values of the RGB primaries

$$\begin{matrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{matrix}$$

The three arrays of n pairs of CCOs define a colour look-up table (LUT) in order to transform non-linear RGB to another RGB specification that is linear with respect to luminous intensity and suitable to be transformed with the 3x3 matrix into the CIEXYZ reference colour space.

$$R' = R\_LUT[R]$$
$$G' = G\_LUT[G]$$
$$B' = B\_LUT[B]$$

The same entries R, B, G must be defined in each array, but not necessarily in the same order. The value n=0 indicates that the colour look-up table is not part of the calibration data parameter.

Elements

When colour model indicator is CMYK, additionally to the white reference value, the calibration data parameter specifies a table of CIEXYZ values for the colours resulting from a grid of specific combinations of C, M, Y, and K colour components. The n grid values XYZ in the CIEXYZ reference colour space are specified for each of the n grid location CMYK. The grid location is specified in terms of % dot of C, M, Y, K.

NOTE — Use of % dot refers to dot size as it will be printed (once dry). Since this % refers to a dry dot, it is equivalent to a % intensity

References:
4.7.7

5.3.X FONT PROPERTIES

Parameters:

list of 4-tuples of (property indicator, property value type, property value,
priority) (n[IX,IX,{I|R|IX|S|OC},I])

where standardized values of the property indicator include:

1: font index (IX)
2: standard version (I)
3: design source (S)
4: font family (S)
5: posture (IX)
6: weight (IX)
7: proportionate width (IX)
8: included glyph collections (nI32)
9: included glyphs (mIX)
10: design size (R)
11: minimum size (R)
12: maximum size (R)
13: design group (3OC)
14: structure (IX)

and standardized values for property value type include:

1: integer
2: real
3: index
4: string
5: octet
6: i32

The property indicator is of data type index; the datatypes in parentheses indicate the property value type associated with the property indicator.

Description:

Font properties are described which may be used for substituting a font specified by property 1, font index, which corresponds to a font name, defined by the FONT LIST element. The property 1, font index, can be used in the FONT INDEX element.

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 could be given priority 10 and all other properties 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 properties which may be referenced by the property indicator are from ISO/IEC 9541. All of the assigned values are from the Minimum Font Description Subset of ISO/IEC 9541-2. Note that the font name itself (referenced by font index), which subsumes all other properties, is one of the properties. Values of property indicator above 14 are reserved for registration and future standardization, and negative values are available for implementation-dependent use.

The priorities 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. Selects a font resource name which resides in the Metafile FONT LIST.

Standard version. Gives the version number of the ISO 9541 which is assumed by the writer of the metafile and the formulator of this font reference. Legal values are: 1, corresponding to ISO 9541 version 199x (first version).

NOTE — ISO 9541 stipulates that its version number will be the year of publication, e.g., the integer 1990 if the first version is published in the year 1990. Because CGM maps 9541 font information into CGM syntax, the ability to use 9541 versions beyondd the first (e.g., 1990) will require amendment to CGM.

Design source. The organizational name of the typeface design source, as specified in ISO 9541.

Font family. The name of the font family, for example *Courier*.

Posture. The posture of a font; assigned values are:

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 values above 6 are reserved for registration and future standardization.

Weight. The font weight is a measure of the 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);

Weights are ordered according to increasing weight. Weight values above 9 are reserved for registration and future standardization.

Proportionate width. The proportionate width is an indication of the relative ratio of character height to character width. Assigned values are:

0: not applicable;
1: ultra condensed (lowest ratio of glyph width to font height);
2: extra condensed;

Elements

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);

Proportionate widths are ordered according to increasing width. Proportionate width values above 9 are reserved for registration and future standardization.

Included glyph collections. A list of glyph collection identifiers registered under ISO 10036. The selected font should have these glyph collections available according to the associated priority.

Included glyphs. A list of character set indices. CGM separates character set (glyph collection) from font. A mechanism exists (GLYPH MAPPING) for associating individual glyph identifiers (for glyphs registered under ISO 10036) with a character set index. This mechanism may be used in conjunction with this element to construct the specification of included glyphs that the font selected to satisfy this font reference should have (according to the associated priority).

Design size. The recommended optimal body size, measured in millimetres, at which the font resource is designed to be used.

Minumum size and maximum size. The recommended minimum and maximum body sizes, measured in millimetres, defining the range over which the font resource is designed to be used.

Design group. The design grouping of the typeface of the font resource consists of three components: the typeface *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 subclasses 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 design 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 properties weight, proportionate width, posture, structure, specify further typographic variations on the design group

Structure. Structure indicates the structure of strokes of the glyph shapes of the font resource. Assigned values are:

0: undefined or not applicable;
1: solid — the shape contains no voids or patterns within the strokes;
2: outline — the shape includes only the outer edges of the strokes.

Structure values above 2 are reserved for registration and future standardization.

References:
4.3.4
4.3.5

5.3.22 GLYPH MAPPING

Parameters:

character set index (IX)

basis set character set type (one of: 94-character G-set, 96-character G-set,
94-character multibyte G-set, 96-character multibyte G-set, complete code) (E)
basis set designation sequence tail (S)
octets per code (I)
list of pairs of (code, glyph name) (n[mOC,I32])

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 is selected by the character set type and designation sequence tail parameters, as described under the CHARACTER SET LIST element (5.3.14). The effect of several GLYPH MAPPING elements with the same character set index is not cumulative. The basis set for the GLYPH MAPPING each time is that specified by the character set type and designation sequence tail parameters. It is not the result of previous GLYPH MAPPING elements. 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 an integer identifier in the range $1..(2^{32}-1)$ which is registered by the ISO Glyph Registration Authority, AFII.

NOTE — Each encoding of this part of this International Standard provides a means to more efficiently represent sequences of pairs which have a uniform increment of 1 in the values both components of successive pairs in the sequence

References:
4.3.4
4.3.5

5.3.23 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.

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

Elements

References:
4.6.12

Page 56

Subclause 5.4.2, first paragraph of description:

Change "red, green, and blue" to "direct"

Page 56

Subclause 5.4.3, changed the enumerated list of line width specification modes from:

(one of: absolute, scaled)

to:

(one of: absolute, scaled, fractional, mm)

Page 56

Subclause 5.4.3, delete the second paragraph of 'Description:' and replace the first by:

One of four methods of specifying geometric aspects associated with lines and the transformation behaviour of those aspects is selected. See clause 4 for a description of the meanings of the four styles.

Page 56

Subclause 5.4.3, add to the References:

4.4.3
4.7.5

Page 56

Subclause 5.4.4, changed the enumerated list of marker size specification modes from:

(one of: absolute, scaled)

to:

(one of: absolute, scaled, fractional, mm)

Page 56

Subclause 5.4.4, delete the second paragraph of 'Description:' and replace the first by:

One of four methods of specifying geometric aspects associated with lines and the transformation behaviour of those aspects is selected. See clause 4 for a description of the meanings of the four styles.

Page 56

Subclause 5.4.3, add to the References:

4.4.3
4.7.5

Page 57

Subclause 5.4.5, changed the enumerated list of edge width specification modes from:

(one of: absolute, scaled)

to:

(one of: absolute, scaled, fractional, mm)

Page 57

Subclause 5.4.5, delete the second paragraph of 'Description:' and replace the first by:

One of four methods of specifying geometric aspects associated with lines and the transformation behaviour of those aspects is selected. See clause 4 for a description of the meanings of the four styles.

Page 57

Subclause 5.4.3, add to the References:

4.4.3
4.7.5

Page 58 (Am.1, page 33, 5.4.11)

Subclause 5.4.11, add after the two current values of *line width specifier*:

if the line width specification mode is 'fractional',
line width as fraction of default viewport (R)

if the line width specification mode is 'mm',
physical line width in millimetres (R)

Page 58 (Am.1, page 34, 5.4.12)

Subclause 5.4.12, add after the two current values of *marker size specifier*:

if the marker size specification mode is 'fractional',
marker size as fraction of default viewport (R)

if the marker size specification mode is 'mm',
physical marker size in millimetres (R)

Elements

Page 58 (Am.1, page 35, 5.4.14)

Subclause 5.4.14, change the enumeratel list of interior styles from:

(one of: hollow, solid, pattern, hatch, empty) (E)

to:

(one of: hollow, solid, pattern, hatch, empty, geometric pattern, interpolated) (E)

Page 58 (Am.1, page 36, 5.4.15)

Subclause 5.4.15, add after the two current values of *edge width specifier*:

if the edge width specification mode is 'fractional',
edge width as fraction of default viewport (R)

if the edge width specification mode is 'mm',
physical edge width in millimetres (R)

Page 61

Add new subclause 5.4.16 defining INTERIOR STYLE SPECIFICATION MODE:

5.4.16 INTERIOR STYLE SPECIFICATION MODE

Parameters:

interior style specification mode (one of: absolute, scaled, fractional, mm) (E)

Description:

One of four methods of specifying geometric aspects associated with lines and the transformation behaviour of those aspects is selected. See clause 4 for a description of the meanings of the four styles.

References:
4.7.5

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:

5.5.13 PROTECTION REGION INDICATOR

Parameters:

region index (IX)
region indicator (IX)

Description:

The region indicator determines how the protection region associated with the given index is used. The region indicator may have the values.

0: off, the region is not used;
1: clip, the region is used for clipping;
2 shield, the region is used for shielding.

It is independent of CLIP INDICATOR, which affects only the use of CLIP RECTANGLE.

References:
4.5.4

5.5.14 GENERALIZED TEXT PATH MODE

Parameters:

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:
4.6.3.3

5.5.15 MITRE LIMIT

Parameters:

mitre limit (R)

Description:

Elements

The mitre limit is defined for subsequent line and edge elements. Miter limit is measured as a scale factor applied to the current line or edge width. See clause 4 for a description of the effect of mitre limit.

References:
4.5.5

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.

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' elements between the BEGIN COMPOUND PATH and END COMPOUND PATH elements specify 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' elements between the BEGIN COMPOUND PATH and END COMPOUND PATH elements specify 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' elements between the BEGIN COMPOUND PATH and END COMPOUND PATH elements specify the path the text string is to follow, and the method of orienting characters along the path is defined by the mode.

Page 69

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 values.

Page 77

Subclause 5.6, add the following Graphical Primitive Elements:

5.6.22 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.

References:
4.6.9

5.6.23 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

Elements

significance of the parameterization and details of rendering of hyperbolic arcs.

References:
4.6.10

5.6.24 NON-UNIFORM B-SPLINE

Parameters:

spline order (I)
number of control points (I)
control points (nP)
list of knots ((n+m)R)
parameter start value (R)
parameter end value (R)

Description:

The spline order must be a positive integer. The knot sequence must form a non-decreasing sequence of numbers. 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 — if the spline order is m (which equals $k+1$, where k is the degree) and the number of control points is n then the number of knots is $(n+m)$.

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 knot 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:
4.6.11.1

5.6.25 NON-UNIFORM RATIONAL B-SPLINE

Parameters:

spline order (I)
number of control points (I)
control points (nP)
list of knots ((n+m)R)
parameter start value (R)
parameter end value (R)
weights (nR)

Description:

The spline order must be a positive integer. The knot sequence must form a non-decreasing sequence of numbers. 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 — if the spline order is m (which equals $k+1$, where k is the degree) and the number of control points is n then the number of knots is $(n+m)$.

The *weights* array contains one real number for each control point in the *control points* array. The meaning of the weights is described in clause 4.

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 knot 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 rational B-spline curve is generated for parameter values between the parameter start value and parameter end value.

References:
4.6.11.1

5.6.26 POLYBEZIER

Parameters:

continuity indicator (IX)
point list (4nP)

Description:

This element defines one or more cubic Bezier curves. The association of points in the parameter list with control points is dependent upon the *continuity indicator* and is defined in clause 4. The derivation of the cubic parametric equations defining the curves is given in clause 4.

The relationship of the Nth Bezier curve to the (N-1)th, if there is more than one curve, is specified by the *continuity indicator*. Valid values are:

1: discontinuous — successive curves may be disjoint;
2: continuous — successive curves are connected, final point of Nth curve to initial of (N+1)th.
3: smooth — successive curves are connected and first derivatives match at the junction of the two curves.

References:
4.6.11.2

5.6.27 SYMBOL

Parameters:

position (P)
symbol index (IX)

Description:

Elements

The symbol corresponding to the symbol index parameter in the symbol library specified by the current SYMBOL LIBRARY INDEX is dimensioned according to SYMBOL SIZE, oriented according to SYMBOL ORIENTATION, and drawn at the specified position point. The symbol is displayed according to the current SYMBOL COLOUR.

References:
4.6.12

5.6.28 BITONAL TILE

Parameters:

tile identifier (I)
compression type (IX)
cell background colour (CO)
cell foreground colour (CO)
compressed colour specifiers (BS)

Description:

The tile identifier will be used as the tiling index.

The *compression type* specifies the compression type used. The following methods are defined:

0: null background
1: null foreground
2: T6
3: T4 1-dimensional
4: T4 2-dimensional
5: bitmap (uncompressed)
6: LZW

Compression types greater than 6 are reserved for registration or future standardization. If the method is T4, the image is encoded according to the one or two dimensional scheme defined by CCITT Recommendation T4 (Group 3 facsimile). If the method is T6 two dimensional scheme defined in CCITT Recommendation T6 (group 4 facsimile). Null background and null foreground indicate that all cells in the tile are known to be background or foreground respectively. In this case the tile has no encoded content. The bitstream parameter is null. The sequence of compressed colour specifiers is compressed according to this compression type parameter and stored as a compressed binary data object in the metafile.

The cell colour specifiers have only two values, the indexes 0 and 1. Index 0 designates the cell background colour. Colour 1 designates the cell foreground colour. The precompressed or uncompressed colour specifiers considered as a binary data stream are represented at 1 bit per cell.

NOTE — Compression method values are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a compression method value has been approved by the ISO Subcommittee for Computer Graphics, the compression method value will be assigned by the Registration Authority.

References:
4.6.5.i
D.4.6

5.6.29 TILE

Parameters:

tile identifier (I)
compression type (IX)
cell colour precision (I)
compressed colour specifiers (BS)

Description:
The tile identifier will be used as the tiling index.

The *compression type* specifies the compression method used. The following methods are defined:

1: T4 1-dimensional
2: T4 2-dimensional
3: T6
4: LZW
5: bitmap (uncompressed)
6: null background
7: null foreground

Compression types greater than 7 are reserved for registration or future standardization. If the method is T4, the image is encoded according to the one or two dimensional scheme defined by CCITT Recommendation T4 (Group 3 facsimile). If the method is T6 two dimensional scheme defined in CCITT Recommendation T6 (group 4 facsimile). Null Background and null foreground indicate that all cells in the tile are known to be background or foreground respectively. In this case the tile has no encoded content. The bitstream parameter is null. The sequence of compressed colour specifiers is compressed according to this compression type parameter and stored as a compressed binary data object in the metafile.

NOTE — The LZW (Lempel, Ziv, Welch) adaptive compression method converts variable-length strings of input characters into fixed-length code words. The LZW encoding is a variation of the Lempel-Ziv technique. The LZW algorithm has gained wide acceptance and is used in many data compression programs. Its wide use can be attributed to its speed and high compression results.

The *cell colour precision* defines the colour precision of the colour specifiers in the pre-compressed or uncompressed datastream. When 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 colour precision is other than 1 and the colour selection mode is not indexed.

NOTE — Compression method values are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a compression method value has been approved by the ISO Subcommittee for Computer Graphics, the compression method value will be assigned by the Registration Authority.

References:
4.6.5.1
D.4.6

Page 89

Subclause 5.7.19, CHARACTER SET INDEX, first line of description, after CHARACTER SET LIST add:

or GLYPH MAPPING

Page 89

Elements

Subclause 5.7.20, ALTERNATE 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:

geometric pattern, interpolated

Page 91

Section 5.7.23, FILL COLOUR, add to the end of the last sentence of the last paragraph:

...or geometric pattern.

Page 92

Section 5.7.25, PATTERN INDEX, append the 4th paragraph to the end of the 2nd paragraph.

Page 92

Section 5.7.25, PATTERN INDEX, insert new 3rd paragraph:

When the PATTERN INDEX ASF is 'individual' and the interior style is 'geometric pattern' subsequent filled-area elements are displayed using this pattern index. The pattern index is a pointer into the table of geometric patterns defined by GEOMETRIC PATTERN.

Page 95

Section 5.7.31, FILL REFERENCE POINT, change paragraph 2, line 1 of 'Description' from:

When the currently selected interior style is 'pattern'...

to:

When the currently selected interior style is 'pattern' or 'geometric pattern'...

Page 95

Section 5.7.31, FILL REFERENCE POINT, replace paragraph 4 with:

When the currently selected interior style is 'interpolated' the FILL REFERENCE POINT provides one of the reference points in the definition of the interior style.

Page 95

Section 5.7.31, FILL REFERENCE POINT, replace paragraph 5 with:

The common origin for the interiors of filled areas means that separate filled areas that have the same hatch index and that abut have a visually continuous interior rendering across all of the filled areas.

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

Subclause 5.7.33, replace the definition of PATTERN SIZE with:

5.7.33 PATTERN SIZE

Parameters:

if the interior style specification mode is 'absolute',
pattern height vector, x component (VDC)
pattern height vector, y component (VDC)
pattern width vector, x component (VDC)
pattern width vector, y component (VDC)

if the interior style specification mode is 'scaled',
pattern vector components (4R)

if the interior style specification mode is 'fractional',
pattern vector components (4R)

if the interior style specification mode is 'mm',
pattern vector components (4R)

Description:

The pattern size is set to the values specified by the parameters.

When the INTERIOR STYLE is set to *pattern* or *geometric pattern* subsequent filled-area elements are displayed using this pattern size. See 4.6 for a list of filled-area elements.

Pattern size is comprised of two vectors, a height vector and a width vector. In the general case the pattern size vectors and the FILL REFERENCE POINT define a parallelogram, the pattern box. When the interior style is *pattern* this pattern box is divided into cells, nx in the width vector direction and ny in the height vector direction, where nx and ny are the colour array dimensions of the pattern table entry selected by the current pattern index. When the interior style is *geometric pattern* the associated pattern extent rectangle is mapped onto the pattern box parallelogram.

The units in which the pattern size vectors are specified, as well as their behaviour and the behaviour of the rendered interior under transformations, is determined by the current value of the INTERIOR STYLE SPECIFICATION MODE. See clause 4.

Elements

When the interior style is *pattern* the array of colours of the current pattern is mapped onto the array of cells as follows. The colour array element (1,ny) is mapped to the pattern box cell which is located at the FILL REFERENCE POINT. Colour array elements with increasing first dimension are associated with successive cells in the direction of the height vector. In this way, each of the nx*ny colour array elements is associatedd with one of the nx*ny cells of the pattern box.

Conceptually, the pattern box so defined is replicated in directions parallel to the vectors of the PATTERN SIZE element until the interior of a filled-area element to which the pattern is to be applied is completely covered. The coincidence of this imposed pattern and the interior to which it is to be applied deefines the interior style for the filled-area element being displayed.

References:
4.6.4
4.7.8
D.4.6

Page 98

Subclause 5.7, add the following attribute elements:

5.7.32 LINE AND EDGE TYPE DEFINITION

Parameters:

line type (IX)
dash cycle 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* is the index of linetype being defined. It must be negative, to avoid conflict with standardized and registered values. The *list of dash elements* 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 l_1 units, gap for l_2 units, solid for l_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 *dash cycle repeat length* defines the length of one complete cycle of the dash pattern. The lengths of the dash elements are normalized so that the sum of the specifiers in the *list of dash elements* equals the *dash cycle repeat length*.

The units of the *dash cycle repeat length* are determined by the value of LINE WIDTH SPECIFICATION MODE. The value of *scaled* indicates that the implementation may normalize and map the sum of the dash pattern elements at its discretion. Otherwise the units to which the dash elements are normalized are as defined by that mode.

References:
4.7.13

5.7.38 LINE CAP

Parameters:

line cap indicator (IX)
dash cap indicator (one of: unspecified, match) (E)

Description:

The line cap and dash cap styles are 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 defined:

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 — 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 — 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 'match' the endpoints of all the dashes have the style defined by the *line cap indicator*. When it is *unspecified* the endpoints of all dashes have the have implementation dependent treatment (as in 8632 version 1), except for the open endpoints of the lines, which have the style defined by the *line cap indicator*.

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:
4.7.13

5.7.39 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 individual line elements as well as the junctions between successive individual line elements in compound line elements. The defined values are:

1: unspecified — as in 8632 version 1, any implementation dependent treatment is acceptable.
2: mitre — the outer edges of the two adjoining line segments are extended until they meet at a point.
3: round — 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 — the adjoining line segments are terminated with a butt cap, and the resulting triangular notch is filled in.

Elements

NOTE — Line join values are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority. When a line join value has been approved by the ISO Subcommittee for Computer Graphics, the line join value will be assigned by the Registration Authority.

References:
4.7.13

5.7.40 LINE TYPE CONTINUATION

Parameters:

continuation mode (IX)

Description:

The behaviour of dashed line patterns at the vertices of individual line elements and the junctions between successive individual line elements in compound 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;
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. For this style the initial and final points of the line are included in those which must be "inked".

Positive values above 4 are reserved for future standardization and registration. The latter may include some very specific requirements from application areas.

NOTE — The styles *restart* and *adaptive continue* are likely to yield poor results if the drawn lines consist of many short segments.

References:
4.7.13

5.7.41 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.

References:
4.7.13

5.7.42 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 'off' or 'on'. The value 'off' indicates that the corresponding score type is not used. The value 'on' indicates that the corresponding score is used.

Any combination of score types may be active simultaneously.

The text score may be changed in Text Open State (TOS).

References:
4.6.3.2
4.7.6

Page 98

Subclause 5.7, add the following to the attribute elements:

5.7.43 RESTRICTED TEXT TYPE

Parameters:

index (IX)

Description:

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:
4.7.6

Elements

5.7.44 HATCH STYLE DEFINITION

Parameters:

hatch index (IX)
style indicator (one of: parallel, crosshatch) (E)
hatch direction vectors

(4VDC) if the INTERIOR STYLE SPECIFICATION MODE is *absolute*

(4R) otherwise

duty cycle length (R)
number of hatch lines (I)
list of gap widths (nI)
list of line types (nIX)

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 *number of hatch lines* defines the number of entries in the arrays of gap widths and line types. The *list of gap widths* defines the gaps between the centres of the lines comprising the hatch. Each gap specification defines the gap following the associated line — that is, the first gap follows the first drawn hatch line.

The colour of each hatch line is the current fill colour. The *list of line types* defines the line type of each the line comprising the hatch.

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 line width of the hatch lines is equal to the duty cycle length divided by the sum of the gap widths — in the units by which the gap widths are specified it is one unit.

The hatch direction vectors specify the directions of the hatch lines. The specification units are determined by the current INTERIOR STYLE SPECIFICATION MODE, as are the transformation properties of the hatch style. See clause 4. Only the first vector is significant if the hatch type is 'parallel'. All hatch lines in the first direction are drawn first, followed by all lines in the second direction (if the style is a crosshatch).

References:
4.7.8

5.7.45 GEOMETRIC PATTERN

Parameters:

geometric pattern index (IX)
segment identifier (N)

pattern extent (2P)

Description:

This element defines a geometric pattern and associates it with an index in the geometric pattern table for future reference. The geometric pattern index parameters defines the index by which the geometric pattern is subsequently referenced with the PATTERN INDEX element. Legal values for the geometric pattern index are positive integers.

The segment identifier specifies the segment which is to be used to define the geometric pattern.

The pattern extent is specified by two points. 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. Valid values for the two points are any two distinct points.

When interior style is *geometric pattern* the interior of a filled-area is filled with the geometric pattern specified by the pattern index and this element. The segment is copied in a manner similar to COPY SEGMENT. The segment is clipped to the rectangle defined by the pattern extent. CLIP INHERITANCE has no effect. The inheritance filter mechanism controls whether attributes are inherited from the current modal values or those specified in the segment are used. All segment attributes except SEGMENT TRANSFORMATION, if present, are ignored. The resulting rectangle is mapped onto the pattern box as described under PATTERN SIZE, and the geometric pattern fills the interior as described under that element.

Redefinition of a geometric pattern index which has already been defined and is in use has no dynamic or retroactive effects. It affects only those primitives coming after the redefinition.

NOTE — The default segment visibility is *visible*. When defining a segment for use as a geometric pattern the segment visibility should be set to *invisible*.

References:
4.7.8
D.4.6

5.7.46 INTERPOLATED INTERIOR DEFINITION

Parameters:

style (IX)
reference geometry

if INTERIOR STYLE SPECIFICATION MODE is *absolute* (2nVDC),

otherwise (2nR)

number of stages (I)
list of stage designators (mR)
list of reference colours (kCO)

Description:

The *style* parameter selects the way of defining the coloured plane. The following values are defined:

Elements

1: parallel;
2: elliptical;
3: triangular.

The geometry of the shaded plane is defined relative to the FILL REFERENCE POINT. The scalars of *geometry definition* are applied as follows:

parallel: the number of scalars must be 2. The pair of scalars are are respectively the x and y offset from the FILL REFERENCE POINT to a second reference point.

elliptical: the number of scalars must be 4. The first pair of scalars are respectively the x and y offset from the FILL REFERENCE POINT to the first CDP of an ellipse and the second pair are respectively the x and y offset from the FILL REFERENCE POINT to the second CDP of the ellipse.

triangular: the number of scalars must be 4. The first pair of scalars are respectively the x and y offset from the FILL REFERENCE POINT to the second corner of a reference triangle and the second pair are respectively the x and y offset from the FILL REFERENCE POINT to the third corner of the reference triangle.

For parallel and elliptical styles, one or more bands of parallel or concentric interpolated colours are defined as follows.

Delimiting points dividing adjacent interpolation stages are defined along the line through the FILL REFERENCE POINT and the first geometry reference point. One delimiter of the first stage (there must be at least one stage) is the FILL REFERENCE POINT. If the line through the FILL REFERENCE POINT and the first geometry reference point is designated L, the distance from the FILL REFERENCE POINT to the first geometry reference point is designated d, and the i-th stage designator is denoted S_i, then addition stage delimiters are defined as follows. The i-th stage delimiter is located on the line L at a distance dS_i from the FILL REFERENCE POINT (positive is in the direction of the first geometry reference point, negative is in the opposite direction).

The colours are assigned to the stage delimiters in order, and linearly interpolated across bands that are defined by adjacent stage delimiters.

For the triangular style, the first reference colour is applied at the FILL REFERENCE POINT, which is the first corner of the interpolated triangle. S_1 is applied along the line from the FILL REFERENCE POINT to the second corner to determine the position of the second colour. S_2 is applied along the line from the FILL REFERENCE POINT to the third corner to determine the position of the second colour. The interpolation is defined on these three colour-tagged points as defined in clause 4.

References:
4.7.8

5.7.47 EDGE CAP

Parameters:

edge cap indicator (IX)
dash cap indicator (one of: unspecified, match) (E)

Description:

The edge cap and dash cap styles are defined for subsequent line elements. The *edge cap indicator* determines the appearance of open endpoints (as opposed to interior vertices) of line elements. The following values are defined:

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 — 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 — 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 dash cap indicator determines the appearance of the endpoints of individual dashes for subsequent dashed edges. When it is 'match' the endpoints of all the dashes have the style defined by the *edge cap indicator*. When it is *unspecified* the endpoints of all dashes have the have implementation dependent treatment (as in 8632 version 1), except for the open endpoints of the edges, which have the style defined by the *edge cap indicator*.

NOTE — Line 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:
4.7.14

5.7.48 EDGE JOIN

Parameters:

edge join indicator (IX)

Description:

The edge join style is defined for subsequent filled-area elements. The edge join style defines the appearance of interior vertices of individual filled-area elements as well as the junctions between successive individual filled-area elements in compoint filled-area elements. The defined values are:

1: unspecified — as in 8632 version 1, any implementation dependent treatment is acceptable.
2: mitre — the outer edges of the two adjoining edge segments are extended until they meet at a point.
3: round — 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 — the adjoining edge segments are terminated with a butt cap, and the resulting triangular notch is filled in.

NOTE — Edge join values are registered in the ISO International Register of Graphical Items, which is maintained by the Registration Authority When a edge join value has been approved by the ISO Subcommittee for Computer Graphics, the edge join value will be assigned by the Registration Authority

References:
4.7.14

5.7.49 EDGE TYPE CONTINUATION

Parameters:

Elements

continuation mode (IX)

Description:

The behaviour of dashed edge patterns at the vertices of individual edges of filled-area elements and the junctions between successive individual filled-area elements in compound filled-area elements is determined. The following standardized values are defined:

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. For this style the initial and final points of the line are included in those which must be "inked".

Positive values above 4 are reserved for future standardization and registration. The latter may include some very specific requirements from application areas.

NOTE — The styles *restart* and *adaptive continue* are likely to yield poor results if the drawn lines consist of many short segments.

References:
4.7.14

5.7.50 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.

References:
4.7.14

5.7.51 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.

References:
4.6.12

5.7.52 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 (CD)

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:
4.6.12
D.4.6

5.7.53 SYMBOL SIZE

Parameters:

scale indicator (one of: height, width, both) (E)
symbol height (VDC)
symbol width (VDC)

Description:

The *symbol height* and *symbol width* are set to the values specified by the parameters. See 4.6 for a list of symbol elements, as well as a description of the use of the symbol attributes by generators and interpreters.

Valid values of *symbol height* and *symbol width* are positive VDC.

References:
4.6.12

Elements

5.7.54 SYMBOL ORIENTATION

Parameters:

symbol up, x component (VDC)
symbol up, y component (VDC)
symbol base, x component (VDC)
symbol base, y component (VDC)

Description:

Two vectors are defined which determine the orientation and skew of symbols in subsequent symbol elements. See 4.6 for a list of symbol elements, as well as 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:
4.6.12

Page 98 (Am.1 page 38)

5.5.11, SAVE PRIMITIVE CONTEXT:

add after LINE CLIPPING MODE:

LINE CAP
LINE JOIN
LINE TYPE CONTINUATION
LINE TYPE INITIAL OFFSET

add after ALTERNATE CHARACTER SET INDEX:

TEXT SCORE TYPE
RESTRICTED TEXT TYPE
GENERALIZED TEXT PATH MODE

add after INTERIOR STYLE:

INTERPOLATED INTERIOR DEFINITION

add after EDGE CLIPPING MODE:

EDGE CAP
EDGE JOIN
EDGE TYPE CONTINUATION
EDGE TYPE INITIAL OFFSET

add after PICK IDENTIFIER:

SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL SIZE

SYMBOL ORIENTATION

add after CLIP RECTANGLE:

PROTECTION REGION INDICATOR

after TRANSPARENCY:

MITRE LIMIT

Page 100 (Am.1 page 21)

5.10.1.2, INHERITANCE FILTER, add to the filter selection list:

add after LINE CLIPPING MODE:

LINE CAP
LINE JOIN
LINE TYPE CONTINUATION
LINE TYPE INITIAL OFFSET

after TEXT COLOUR:

TEXT SCORE TYPE
RESTRICTED TEXT TYPE

after TEXT ALIGNMENT:

GENERALIZED TEXT PATH MODE

after PATTERN INDEX:

INTERPOLATED INTERIOR DEFINITION

after EDGE CLIPPING MODE:

EDGE CAP
EDGE JOIN
EDGE TYPE CONTINUATION
EDGE TYPE INITIAL OFFSET

after PICK IDENTIFIER:

SYMBOL LIBRARY INDEX
SYMBOL COLOUR
SYMBOL SIZE
SYMBOL ORIENTATION

after TRANSPARENCY:

MITRE LIMIT

Page 100 (Am.1 page 42)

Elements

5.10.1.3, CLIP INHERITANCE, add a sentence at the start of the clause.

The description of clipping in this subclause also applies to the protection region used for clipping and shielding of arbitrary areas.

Page 100

Add a new subclause after 5.10.2.4:

5.10.2.5 SEGMENT VISIBILITY

Parameters:

segment identifier (N)
visibility (one of: visible, invisible) (E)

Description:
When the visibility attributes is set to *visible* the segment is displayed. When the attribute is set to *invisible* the segment is not displayed.

References:
4.12.1

6 Defaults

Page 101

Page 101

Clause 6, add to the definition of the LINE WIDTH default:

if LINE WIDTH SPECIFICATION MODE is 'fractional', 0.001; if LINE WIDTH SPECIFICATION MODE is 'mm', 0.35;

Page 101

Clause 6, add to the definition of the MARKER SIZE default:

if MARKER SIZE SPECIFICATION MODE is 'fractional', 0.001; if MARKER SIZE SPECIFICATION MODE is 'mm', 0.35;

Page 101

Clause 6, add to the definition of the EDGE WIDTH default:

if EDGE WIDTH SPECIFICATION MODE is 'fractional', 0.001; if EDGE WIDTH SPECIFICATION MODE is 'mm', 0.35;

Page 103

Clause 6, change the default of PATTERN SIZE to:

0,dy,dx,0, where depending upon the value of INTERIOR
STYLE SPECIFICATION MODE dy and dx are respectively:

— height and width of default VDC extent if *absolute*;
— height and width of some device-dependent "nominal" if *scaled*;
— 0.,1.,1.,0. if *fractional*;
— 0.,1.,1.,0. if *mm*;

Page 101

Clause 6: Add the following default specifications:

Protection region:	default VDC Extent
COLOUR MODEL:	1 (RGB)
COLOUR CALIBRATION:	reference white value, D65 0.391 0.365 0.192

Defaults

	0.212 0.701 0.087 0.019 0.112 0.958 0 (no lookup table)
INTERIOR STYLE SPECIFICATION MODE:	absolute
PROTECTION REGION INDICATOR:	0 (off)
MITRE LIMIT:	1.0
GENERALIZED TEXT PATH MODE:	off
LINE CAP:	1 (unspecified, as 8632 version 1)
LINE JOIN:	1 (unspecified, as 8632 version 1)
LINE TYPE CONTINUATION:	1 (unspecified, as 8632 version 1)
LINE TYPE INITIAL OFFSET:	0.0
TEXT SCORE TYPE:	all text scores are 'off'
RESTRICTED TEXT TYPE:	1 (basic)
GEOMETRIC PATTERN	1
INTERPOLATED INTERIOR DEFINITION:	style, 'parallel' reference points — VDC extent reference colours — device-dependent background colour if COLOUR SELECTION MODE is 'direct', 0 if COLOUR SELECTION MODE is 'indexed'
EDGE CAP:	1 (unspecified, as 8632 version 1)
EDGE JOIN:	1 (unspecified, as 8632 version 1)
EDGE TYPE CONTINUATION:	1 (unspecified, as 8632 version 1)
EDGE TYPE INITIAL OFFSET:	0.0
SYMBOL LIBRARY INDEX:	n/a
SYMBOL LIBRARY LIST:	n/a
SYMBOL ORIENTATION	as default CHARACTER ORIENTATION
SYMBOL SIZE	scaling indicator, 'height' height and width, 0.01 of longest side of default VDC Extent.
SEGMENT VISIBILITY	visible

7 Annex D

Page 132

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.

Tile Array

If the number of tiles present does not match the count specified by the BEGIN TILE ARRAY parameter, it is recommended that the missing tiles be treated as encoded as "null background".

GEOMETRIC PATTERN

If an index has not been defined, or a segment has not been defined an empty segment is used for the geometric pattern. The effect is the same as interior style 'empty'.

Page 133

Add to the table in D.5

GEOMETRIC PATTERN TABLE 1, interpreter-dependent.

8 Annex I

Page 144 (Am.1 page 93)

Insert new annex I defining the formal grammar of version 3 metafiles:

Annex I

Formal Grammar of the functional specification of version 3 metafiles.

(This annex forms an integral part of the standard.)

I.1 Introduction

This grammar is a formal definition of a standard CGM extended syntax for Version 3 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.

I.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

I.3 Detailed Grammar

I.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>
	|	<segment>
<picture element>	::=	<control element>
	|	<graphical element>
	|	<closed figure>
	|	<primitive attribute element>
	|	<pattern table element>
	|	<colour table element>
	|	<specification element>
	|	<segment control element>
	|	<compound path>
	|	<protection region>
	|	<tile array>
	|	<extra element>
<segment>	::=	<BEGIN SEGMENT>
		<segment identifier>
		<segment attribute element>*
		<eligible picture element>*
		<compound path>*
		<protection region>*
		<END SEGMENT>
<segment identifier>	::=	<name>
<eligible picture element>	::=	<control element>
	|	<graphical element>
	|	<closed figure>
	|	<primitive attribute element>
	|	<specification element>
	|	<segment control element>
	|	<extra element>
<compound path>	::=	<BEGIN COMPOUND PATH>
		<path type>
		<eligible element within compound path>*
		<END COMPOUND PATH>

<eligible element within

Annex I

compound path> ::= <POLYLINE>
| <DISJOINT POLYLINE>
| <GDP>
| <CIRCULAR ARC 3 POINT>
| <CIRCULAR ARC CENTRE>
| <ELLIPTICAL ARC>
| <CIRCULAR ARC CENTRE REVERSED>
| <HYPERBOLIC ARC>
| <PARABOLIC ARC>
| <NON-UNIFORM B-SPLINE>
| <NON-UNIFORM RATIONAL B-SPLINE>
| <POLYBEZIER>
| <ESCAPE>
| <MESSAGE>
| <APPLICATION DATA>

<path type> ::= <index>

<protection region> ::= <BEGIN PROTECTION REGION>
<region index>
<eligible element within protection region>*
<END PROTECTION REGION>

<region index> ::= <index>

<eligible element within
protection 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>
| <ELLIPSE>
| <ELLIPTICAL ARC>
| <ELLIPTICAL ARC CLOSE>
| <CIRCULAR ARC CENTRE REVERSED>
| <PARABOLIC ARC>
| <HYPERBOLIC ARC>
| <NON-UNIFORM B-SPLINE>
| <NON-UNIFORM RATIONAL B-SPLINE>
| <POLYBEZIER>
| <ESCAPE>
| <MESSAGE>
| <APPLICATION DATA>

<tile array> ::= <BEGIN TILE ARRAY>
<position>
<cell path direction enumerated>
<line progression direction enumerated>
<number of tiles in path direction>
<number of tiles in line direction>

<number of cells/tile in path direction>
<number of cells/tile in line direction>
<cell size in path direction>
<line size in path direction>
<image offset in path direction>
<image offset in line direction>
<number of cells in path direction>
<number of cells in line direction>
<eligible element within tile array>*
<END TILE ARRAY>

<position> ::= <point>

<cell path direction enumerated> ::= <0 DEGREES>
| <90 DEGREES>
| <180 DEGREES>
| <270 DEGREES>

<line progression direction enumerated> ::= <90 DEGREES>
| <270 DEGREES>

<number of tiles in path direction> ::= <integer>

<number of tiles in line direction> ::= <integer>

<number of cells/tile in path direction> ::= <integer>

<number of cells/tile in line direction> ::= <integer>

<cell size in path direction> ::= <real>

<line size in path direction> ::= <real>

<image offset in path direction> ::= <integer>

<image offset in line direction> ::= <integer>

<number of cells in path direction> ::= <integer>

<number of cells in line direction> ::= <integer>

<eligible element within tile array> ::= <BITONAL TILE>
| <TILE>
| <extra element>

I.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>
                                   |   >VERSION 3 SET>

<optional descriptor element>      ::= <description>
                                   |   <VDC TYPE>
                                         <vdc type enumerated>
                                   |   <MAXIMUM COLOUR INDEX>
                                         <colour index>
                                   |   <COLOUR VALUE EXTENT>
                                         <colour value mapping specifier>
                                   |   <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>
                                         <minimum extent>
                                         <maximum extent>
                                   |   &lt;segment&gt;
                                   |   <COLOUR MODEL>
                                         <index>
                                   |   <COLOUR CALIBRATION>
                                         <reference white>
                                         <calibration data>
                                   |   <FONT PROPERTIES>
                                         <font property 4-tuple>+
                                   |   <GLYPH MAPPING>
                                         <index>
```

```
                                <character set type enumerated>
                                  <designation sequence>
                                  <octets per code>
                                  <code glyph name pair>+
                              | <SYMBOL LIBRARY LIST>
                                  <symbol library names>+
                              | <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>
```

Annex I

		<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>
<colour value mapping specifier>	::=	<colour direct>(2)
	|	(<colour scale> <colour offset>)(3)
<colour scale>	::=	<real>
<colour offset>	::=	<real>
<font property 4-tuple>	::=	<index>(2)
		<property value>
		<integer>
<property value>	::=	<integer>
	|	<real>
	|	<index>
	|	<string>
	|	<octet>
<octets per code>	::=	<integer> {n>0}
<code glyph name pair>	::=	<code>
		<glyph name>
		<repeat number>
<code>	::=	<octet>(n)
<glyph name>	::=	<integer>
<repeat number>	::=	<integer>
<reference white>	::=	<real>(3)
<calibration data>	::=	<rgb calibration data>
	|	<cmyk calibration data>
<rgb calibration data>	::=	<3x3 matrix of reals>
		<integer>
		<lookup table entry for red>(integer)
		<lookup table entry for green>(integer)
		<lookup table entry for blue>(integer)
<cmyk calibration data>	::=	<integer>
		<cmyk grid location>(integer)
		<ciexyz grid location>(integer)

```
<lookup table entry for red>      ::= <cco value>(2)

<lookup table entry for green>    ::= <cco value>(2)

<lookup table entry for blue>     ::= <cco value>(2)

<cmyk grid location>              ::= <colour direct>

<ciexyz grid location>            ::= <real>(3)

<symbol library name>             ::= <string>
```

I.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>
                                          <colour direct>
                                    | <specification element>
                                    | <representation element>
                                    | <pattern table element>
                                    | <colour table element>
                                    | <extra element>

<colour>                          ::= <colour index>
                                    | <colour direct>

<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>
                                    | <INTERIOR STYLE 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>
                                  | <FRACTIONAL>
                                  | <MILLIMETRES>

<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>
| <GEOMETRIC PATTERN>
| <INTERPOLATED>

I.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>
| <PROTECTION REGION INDICATOR>
<region index>
<region indicator>
| <GENERALIZED TEXT PATH MODE>
<text path>
| <MITRE LIMIT>
<real>

<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>

<region index>                ::= <index>

<region indicator>            ::= <index>

<text path enumerated>        ::= <OFF>
                                | <NON-TANGENTIAL>
                                | <AXIS-TANGENTIAL>
```

I.3.5 Graphical elements

```
<graphical element>           ::= <polypoint element>
                                | <text element>
                                | <cell element>
                                | <gdp element>
                                | <rectangle element>
                                | <circular element>
                                | <elliptical element>
                                | <pointless element>
                                | <curve element>
                                | <symbol element>
                                | <tile 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>
                                | <POLYBEZIER>
                                    <index>
                                    <control point list>

<point list>                  ::= <point>*

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

```
<control point>                 ::= <point>(4)

<control point list>            ::= <control point>*

<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>

<curve element>                 ::= <HYPERBOLIC ARC>
                                    <point>(3)
                                    <vdc value>(4)
                                  | <PARABOLIC ARC>
                                    <point>(3)
                                  | <NON-UNIFORM B-SPLINE>
                                    <spline order>
                                    <number of control points>
                                    <control points>
```

```
                                      <list of knots>
                                        <parameter start value>
                                        <parameter end value>
                                    | <NON-UNIFORM RATIONAL B-SPLINE>
                                        <spline order>
                                        <number of control points>
                                        <control points>
                                        <list of knots>
                                        <parameter start value>
                                        <parameter end value>
                                        <weights>
                                    | <POLYBEZIER>
                                        <index>
                                        <point>(n)

<spline order>                    ::= <positive integer>  {m}

<number of control points>        ::= <integer>  {n≥m}

<control points>                  ::= <point>(n)

<list of knots>                   ::= <real>(m+n)

<parameter start value>           ::= <real>

<parameter end value>             ::= <real>

<weights>                         ::= <real>(m)

<symbol element>                  ::=   <SYMBOL>
                                        <point>
                                        <index>

<tile element>                    ::= <BITONAL TILE>
                                        <tile identifier>
                                        <compression type>
                                        <cell colour precision>
                                        <cell background colour>
                                        <cell foreground colour>
                                        <compressed colour specifiers>
                                    | <TILE>
                                        <tile identifier>
                                        <compression type>
                                        <cell colour precision>
                                        <compressed colour specifiers>

<tile identifier>                 ::= <index>

<compression type>                ::= <index>
```

I.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>
                                      | <symbol attribute element>

<line attribute element>          ::= <LINE BUNDLE INDEX>
                                        <positive index>
                                      | <LINE TYPE>
                                        <index>
                                      | <LINE WIDTH>
                                        <size value>
                                      | <LINE COLOUR>
                                        <colour>
                                      | <LINE AND EDGE TYPE DEFINITION>
                                        <index>
                                        <real>
                                        <integer>+
                                      | <LINE CAP>
                                        <index>
                                        <dash cap indicator enumerated>
                                      | <LINE JOIN>
                                        <index>
                                      | <LINE TYPE CONTINUATION>
                                        <index>
                                      | <LINE TYPE INITIAL OFFSET>
                                        <real>

<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 SCORE TYPE> <score type>+
	|	<RESTRICTED TEXT TYPE> <index>
<score type>	::=	<index> <score indicator enumerated>
<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>

<dash cap indicator enumerated>   ::= <UNSPECIFIED>
                                      | <MATCH>

<score indicator enumerated>      ::= <RIGHT SCORE>
                                      | <LEFT SCORE>
                                      | <THROUGH SCORE>
                                      | <KENDOT>

<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) | <real>(4)>
                                      | <HATCH STYLE DEFINITION>
                                          <index>
                                          <style indicator enumerated>
                                          <<vdc>(4) | <real>(4)>
                                          <real>
                                          <number of hatch lines>
                                          <gap widths>
                                          <line types>
                                      | <GEOMETRIC PATTERN>
                                          <index>
                                          &lt;segment identifier&gt;
                                          <point pair>
                                      | <INTERPOLATED INTERIOR DEFINITION>
                                          <index>
                                          <geometry definition>
                                          <number of stages>
                                          <stage designators>
                                          <reference colour list>
                                      | <EDGE CAP>
                                          <index>
```

```
                                          <dash cap indicator enumerated>
                                        | <EDGE JOIN>
                                            <index>
                                        | <EDGE TYPE CONTINUATION>
                                            <index>
                                        | <EDGE TYPE INITIAL OFFSET>
                                            <real>

<style indicator enumerated>        ::=   <PARALLEL>
                                        | <CROSSHATCH>

<number of hatch lines>             ::=   <positive integer>   {n>0}

<gap widths>                        ::=   <positive integer>(n)

<line types>                        ::=   <index>(n)

<geometry definition>               ::=   <VDC>(2n)
                                        | <real>(2n)

<stage designators>                 ::=   <real>(m)

<reference colour list>             ::=   <CO>(k)

<colour table element>              ::=   <COLOUR TABLE>
                                            <starting index>
                                            <colour direct>+

<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>
<symbol attribute element>	::= <SYMBOL LIBRARY INDEX>
	<index>
	<SYMBOL COLOUR>
	<colour>
	<SYMBOL SIZE>
	<scale enumerated>
	<vdc>(2)
	<SYMBOL ORIENTATION>
	<vdc>(4)
<scale enumerated>	::= <HEIGHT>
	| <WIDTH>
	| <BOTH>

I.3.7 Closed figure element

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

I.3.8 Escape elements

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

<identifier> ::= <integer>

I.3.9 External elements

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

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

I.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>
| <SEGMENT VISIBILITY>
<segment identifier>
<segment visibility enumerated>

<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>
| <LINE CAP>
| <LINE JOIN>
| <LINE MITRE LIMIT>
| <LINE TYPE CONTINUATION>
| <LINE TYPE INITIAL OFFSET>
| <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>
| <TEXT SCORE TYPE>
| <RESTRICTED TEXT TYPE>
| <CHARACTER HEIGHT>
| <CHARACTER ORIENTATION>
| <TEXT PATH>
| <TEXT ALIGNMENT>
| <GENERALIZED TEXT PATH MODE>
| <FILL BUNDLE INDEX>
| <INTERIOR STYLE>
| <FILL COLOUR>
| <HATCH INDEX>
| <PATTERN INDEX>
| <INTERPOLATED INTERIOR DEFINITION>

```
                                 | <EDGE BUNDLE INDEX>
                                 | <EDGE TYPE>
                                 | <EDGE WIDTH>
                                 | <EDGE COLOUR>
                                 | <EDGE VISIBILITY>
                                 | <EDGE CLIPPING MODE>
                                 | <EDGE CAP>
                                 | <EDGE JOIN>
                                 | <EDGE TYPE CONTINUATION>
                                 | <EDGE TYPE INITIAL OFFSET>
                                 | <FILL REFERENCE POINT>
                                 | <PATTERN SIZE>
                                 | <AUXILIARY COLOUR
                                 | <TRANSPARENCY>
                                 | <SYMBOL LIBRARY INDEX>
                                 | <SYMBOL COLOUR>
                                 | <SYMBOL SIZE>
                                 | <SYMBOL ORIENTATION>

<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>
                                 | &lt;SEGMENT&gt;

<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 visibility enumerated>	::=	<VISIBLE>
	|	<INVISIBLE>
<segment display priority>	::=	<integer>
<segment pick priority>	::=	<integer>

I.4 Terminal symbols

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

<element name>
<integer>
<real>
<vdc value>
<string>
<colour index>
<colour direct>
<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>
<3 x 3 matrix of reals>

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>
<0>
<90>
<180>
<270>
<FRACTIONAL>
<MM>
<NON-TANGENTIAL>
<AXIS-TANGENTIAL>
<PARALLEL>
<CROSSHATCH>
<UNSPECIFIED>
<MATCH>
<HEIGHT>
<WIDTH>
<BOTH>

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

| <BEGIN COMPOUND PATH>
| <END COMPOUND PATH>
| <BEGIN PROTECTION REGION>
| <END PROTECTION REGION>
| <BEGIN TILE ARRAY>
| <END TILE ARRAY>
| <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>
| <COLOUR MODEL>
| <COLOUR CALIBRATION>
| <FONT PROPERTIES>
| <GLYPH MAPPING>
| <SYMBOL LIBRARY LIST>
| <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>
| <INTERIOR STYLE SPECIFICATION MODE>
| <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>
| <PROTECTION REGION INDICATOR>
| <MITRE LIMIT>
| <GENERALIZED TEXT PATH MODE>
| <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>
| <HYPERBOLIC ARC>
| <PARABOLIC ARC>
| <NON-UNIFORM B-SPLINE>
| <NON-UNIFORM RATIONAL B-SPLINE>
| <POLYBEZIER>
| <SYMBOL>
| <BITONAL TILE>
| <TILE>
| <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>
| <LINE AND EDGE TYPE DEFINITION>
| <LINE CAP>
| <LINE JOIN>
| <LINE TYPE CONTINUATION>
| <LINE TYPE INITIAL OFFSET>
| <TEXT SCORE TYPE>
| <RESTRICTED TEXT TYPE>
| <HATCH STYLE DEFINITION>
| <GEOMETRIC PATTERN>
| <INTERPOLATED INTERIOR DEFINITION>
| <EDGE CAP>
| <EDGE JOIN>
| <EDGE TYPE CONTINUATION>
| <EDGE TYPE INITIAL OFFSET>
| <SYMBOL LIBRARY INDEX>
| <SYMBOL COLOUR>
| <SYMBOL SIZE>
| <SYMBOL ORIENTATION>
| <SEGMENT VISIBILITY>

9 Annex J

Page 144 (Am.1 page 93)

Insert new annex J defining conversions between colour models:

Annex J

Conversions between the CIEXYZ reference colour space and the colour spaces defined in the COLOUR MODEL element.

(This annex forms an integral part of the standard.)

J.1 CIELUV

J.1.1 Conversion from the CIEXYZ reference colour space to CIELUV

The CIELUV colour space is related to the CIEXYZ reference colour space by the following equations:

$$L^* = \begin{cases} 116(Y/Y_n)^{\frac{1}{3}} - 16 & \text{for } Y/Y_n > 0.008856 \\ 903.3(Y/Y_n) & \text{for } Y/Y_n \leq 0.008856 \end{cases}$$

$$u^* = 13L^*(u - u_n)$$

$$v^* = 13L^*(v - v_n)$$

with:

$$u = 4X/(X + 15Y + 3Z)$$
$$v = 9Y/(X + 15Y + 3Z)$$
$$u_n = 4X_n/(X_n + 15Y_n + 3Z_n)$$
$$v_n = 9Y_n/(X_n + 15Y_n + 3Z_n)$$

Where X, Y, Z describe the colour stimulus considered, and X_n, Y_n, Z_n are the tristimulus values of the nominal white stimulus (reference white) which according to CIE recommendation is the perfect white reflecting or transmitting diffuser There may be particular applications where the perfect diffuser is not the best choice for the reference white. One example is reflective papers where the reference white must be defined by a contribution from the substrate as well as the illuminant.

Values of X_n, Y_n, Z_n, u_n, and v_n for CIE Standard Illuminant D_{50}, the 2° Standard Observer, and the perfect diffuser are given below. The colorimetric parameters for the reference white will depend on the wavelength range and interval of summation. The default values are based on a range of 380 nm to 700 nm in 10 nm intervals. If a different range and/or interval is used, the colorimetric values must be recalculated.

J.1.2 Conversion from CIELUV to the CIEXYZ reference colour space

The CIEXYZ reference colour space is related to the CIELUV colour space by the following equations:

$$Y = \begin{cases} Y_n \left[\dfrac{L^* + 16}{116}\right]^3 & \text{for } L^* > 8 \\ Y_n \left[\dfrac{L^*}{903.3}\right] & \text{for } L^* \leq 8 \end{cases}$$

$$u = u_n + \frac{u^*}{13L^*}$$

$$v = v_n + \frac{v^*}{13L^*}$$

The x and y chromaticity coordinates can be derived from the u' and v' coordinates by the following equations:

$$x = 9u/(6u - 16v + 12)$$

$$y = 4u/(6u - 16v + 12)$$

The CIEXYZ reference colour space values are then given by:

$$X = x(Y/y)$$

$$Z = (1 - x - y)(Y/y)$$

J.2 CIELAB

J.2.1 Conversion from the CIEXYZ reference colour space to CIELAB

The CIELAB colour space is related to the CIEXYZ reference colour space by the following equations:

$$L^* = \begin{cases} 116(Y/Y_n)^{\frac{1}{3}} - 16, & \text{for } Y/Y_n > 0.008856 \\ 903.3(Y/Y_n), & \text{for } Y/Y_n \leq 0.008856 \end{cases}$$

$$a^* = 500(f(X/X_n) - f(Y/Y_n))$$

$$b^* = 200(f(Y/Y_n) - f(Z/Z_n))$$

where

$$f(C) = \begin{cases} C^{\frac{1}{3}}, & \text{for } C > 0.008856 \\ 7.787C + 16/116, & \text{for } C \leq 0.008856 \end{cases}$$

For the values of X_n, Y_n, Z_n for CIE Standard Illuminant D_{50} see J.1.1.

J.2.2 Conversion from CIELAB to the CIEXYZ reference colour space

The CIEXYZ reference colour space is related to the CIELAB colour space by the following equations:

$$X = X_n \left[\frac{L^* + 16}{116} + \frac{a^*}{500} \right]^3$$

$$Y = \begin{cases} Y_n \left[\frac{L^* + 16}{116} \right]^3, & \text{for } L^* > 8 \\ Y_n \left[\frac{L^*}{903.3} \right], & \text{for } L^* \leq 8 \end{cases}$$

$$Z = Z_n \left[\frac{L^* + 16}{116} - \frac{b^*}{200} \right]^3$$

J.3 RGB

J.3.1 Conversion from the CIEXYZ reference colour space to RGB

The RGB colour space is related to the CIEXYZ reference colour space by the following equations:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} K_1 & K_2 & K_3 \\ K_4 & K_5 & K_6 \\ K_7 & K_8 & K_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$= \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

where $[X_r, Y_r, Z_r]$, $[X_g, Y_g, Z_g]$, and $[X_b, Y_b, Z_b]$ are tristimulus values of the primaries.

The elements of the matrix K are related to the tristimulus values of the primaries as follows:

$$\begin{aligned} K_1 &= (Y_g Z_b - Y_b Z_g)/K_D \\ K_2 &= (X_b Z_g - X_g Z_b)/K_D \\ K_3 &= (X_g Y_b - X_b Y_g)/K_D \\ K_4 &= (Y_b Z_r - Y_r Z_b)/K_D \\ K_5 &= (X_r Z_b - X_b Z_r)/K_D \\ K_6 &= (X_b Y_r - X_r Y_b)/K_D \\ K_7 &= (Y_r Z_g - Y_g Z_r)/K_D \\ K_8 &= (X_g Z_r - X_r Z_g)/K_D \\ K_9 &= (X_r Y_g - X_g Y_r)/K_D \end{aligned}$$

$$K_D = Det(K) = X_r(Y_g Z_b - Y_b Z_g) + X_g(Y_b Z_r - Y_r Z_b) + X_b(Y_r Z_g - Y_g Z_r)$$

The default values of the elements of the matrix K are those specified by SMPTE 'C' RP145.

$$\begin{bmatrix} 3.497 & -1.734 & -0.543 \\ -1.065 & 1.975 & 0.034 \\ -0.055 & -0.197 & 1.051 \end{bmatrix}$$

Analogous to CIELUV, see J.1.1, the matrix element values depend on the wavelength range and interval.

The reference white specified by this transformation through

$X_n = X_r + X_g + X_b$

$Y_n = Y_r + Y_g + Y_b$

$Z_n = Z_r + Z_g + Z_b$

is CIE Standard Illuminant D_{65}, ($X_n = 0.9504$, $Y_n = 1.0000$, $Z_n = 1.0889$).

NOTE — This is not consistent with the illuminant specified for reflections; however it is aligned with current standards and practices for video displays and TV monitors.

In case the initial data are in the form of the CIE chromaticities of the primaries, and the tristimulus values of the reference white X_n, Y_n, and Z_n the following relations hold:

$$K_i = k_i/T_r \ , \ i = 1,2,3$$
$$K_i = k_i/T_g \ , \ i = 4,5,6$$
$$K_i = k_i/T_b \ , \ i = 7,8,9$$

where

$$T_r = k_1X_n + k_2Y_n + k_3Z_n$$
$$T_g = k_4X_n + k_5Y_n + k_6Z_n$$
$$T_b = k_7X_n + k_8Y_n + k_9Z_n$$

and the k_i and k_D are determined by replacing the tristimulus values in the definitions of the K_i and K_D by the corresponding chromaticities.

J.3.2 Conversion from RGB to the CIEXYZ reference colour space

The CIEXYZ reference colour space is related to the RGB colour space by the following equations:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The default values are those specified by SMPTE 'C' RP145

$$\begin{bmatrix} 0.394 & 0.365 & 0.192 \\ 0.212 & 0.701 & 0.087 \\ 0.019 & 0.112 & 0.958 \end{bmatrix}$$

The reference white specified by this transformation is CIE Standard Illuminant D_{65} ($X_n = 0.9504$, $Y_n = 1.0000$, $Z_n = 1.0889$).

NOTE — This is not consistent with the illuminant specified for reflections; however it is aligned with current standards and practices for video displays and TV monitors.

In case the initial data are in the form of the CIE chromaticities of the primaries, and the tristimulus values of the reference white X_n, Y_n, and Z_n the following relations hold:

$$X_i = x_i T_i \ , \ i = r,g,b$$
$$Y_i = y_i T_i \ , \ i = r,g,b$$
$$Z_i = z_i T_b \ , \ i = r,g,b$$

where T_i have been defined in J.3.1.

J.4 CMYK

J.4.1 Conversion from CMYK to the CIEXYZ reference colour space

The conversion of CMYK to the CIEXYZ reference colour space is specified by a set of colour values in the reference space measured on a grid in CMYK space.

The interpreter of a CGM is expected to use interpolation for CMYK values not on the grid. The interpolation methods, such as quadralinear or higher order interpolation, are not standardized.

J.4.2 CMYK Calibration data

Because no calibration data are available for a standard ink set, the COLOUR CALIBRATION element should be present when the colour model indicator is CMYK. As long as no reference colour calibration table is available, the CGM generator should use CMYK colour calibration data based on the specific printing ink, paper, and illumination conditions.

NOTE — Colorimetric definitions are available for standard ink sets, ISO 2846, but are not sufficient to define the CMYK calibration data

ISO/IEC 8632-2:1987/Am.3:1991

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer of Picture Description Information

Part 2

Character Encoding

CONTENTS

5 Method of Encoding Opcodes 1
6 Method of encoding parameters 3
8 Representation of each element 6

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 COMPOUND PATH opcode	3/0	2/12	03/0	02/12
END COMPOUND PATH opcode	3/0	2/13	03/0	02/13
BEGIN PROTECTION REGION opcode	3/0	2/14	03/0	02/14
END PROTECTION REGION opcode	3/0	2/15	03/0	02/15
BEGIN TILE ARRAY opcode	3/0	3/0	03/0	03/0
END TILE ARRAY opcode	3/0	3/1	03/0	03/1
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
INTERIOR STYLE SPECIFICATION MODE opcode	3/2	2/15	03/2	02/15
PROTECTION REGION INDICATOR opcode	3/3	3/0	03/3	03/0
GENERALIZED TEXT PATH MODE opcode	3/3	3/1	03/3	03/1
MITRE LIMIT opcode	3/3	3/2	03/3	03/2
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
NON-UNIFORM RATIONAL B-SPLINE opcode	3/4	2/13	03/4	02/13
POLYBEZIER opcode	3/4	2/14	03/4	02/14
SYMBOL opcode	3/4	2/15	03/4	02/15
BITONAL TILE opcode	3/4	3/0	03/4	03/0
TILE opcode	3/4	3/1	03/4	03/1
LINE AND EDGE TYPE DEFINITION opcode	3/5	2/8	03/5	02/8
LINE CAP opcode	3/5	2/9	03/5	02/9
LINE JOIN opcode	3/5	2/10	03/5	02/10
LINE TYPE CONTINUATION opcode	3/5	2/11	03/5	02/11
LINE TYPE INITIAL OFFSET opcode	3/5	2/12	03/5	02/12
TEXT SCORE TYPE opcode	3/5	2/13	03/5	02/13
RESTRICTED TEXT TYPE opcode	3/5	2/14	03/5	02/14
HATCH STYLE DEFINITION opcode	3/5	2/15	03/5	02/15
GEOMETRIC PATTERN opcode	3/6	2/13	03/6	02/13
INTERPOLATED INTERIOR DEFINITION opcode	3/6	2/14	03/6	02/14
EDGE CAP opcode	3/6	2/15	03/6	02/15
EDGE JOIN opcode	3/6	3/3	03/6	03/3
EDGE TYPE CONTINUATION opcode	3/6	3/4	03/6	03/4
EDGE TYPE INITIAL OFFSET opcode	3/6	3/5	03/6	03/5
SYMBOL LIBRARY INDEX opcode	3/6	3/6	03/6	03/6
SYMBOL COLOUR opcode	3/6	3/7	03/6	03/7
SYMBOL SIZE opcode	3/6	3/8	03/6	03/8
SYMBOL ORIENTATION opcode	3/6	3/9	03/6	03/9
SEGMENT VISIBILITY opcode	3/8	2/12	03/8	02/12

6 Method of encoding parameters

Page 13

Subclause 6.1, add to the list at the bottom of the page:

g) octets (OC);

h) fixed-precision 32-bit unsigned integers (I32).

Page 14

Subclause 6.2, add to the list in the middle of the page:

d) the compressed bitstream (BS) datatype (see 6.15).

Page 14

Subclause 6.2, add to the dash list at the end of the section:

—for compressed binary colour specifiers of Tile Array elements, the SOS/ST delimiters that are also used for String operands are used to demark the Bitstream operand.

Page 22

Sub-clause 6.7, first paragraph, change:

RGB parameters

to:

direct colour parameters

Page 22:

Sub-clause 6.7, third paragraph, first sentence, change:

RGB parameters

to:

direct colour parameters

Page 22

Sub-clause 6.7, third paragraph, change:

representing the red, green and blue colour values to:

representing the direct colour values

Page 22

Sub-clause 6.7, third paragraph, second last sentence, change:

If this value is set to N bits, there are 3N or 4N colour value bits (as determined by the COLOUR MODEL element); i.e. there are only as many bytes in a direct colour parameter as are necessary to hold those 3N or 4N bits.

Page 22

Sub-clause 6.7, third paragraph, change the last sentence to read:

For example, consider RGB colour space with COLOUR PRECISION set to 5 bits. The RGB parameters then have the following form:

Page 22

Sub-clause 6.7, add the following at the end:

Direct colour parameters in other 3-tuple and 4-tuple spaces are encoded correspondingly.

Page 23

Sub-clause 6.8.1, second paragraph, change:

RGB value

to:

direct colour value

Page 23

Sub-clause 6.8.2, second paragraph, change:

RGB bitstream

to:

direct colour bitstream

Pages 23-24

Sub-clause 6.8.3, second paragraph, change:

RGB bitstream

Method of encoding parameters

to:

direct colour bitstream

Page 27 (Am.1, page 2, subclauses 6.13 and 6.14)

Add new subclauses 6.15 and 6.16

6.15 Compressed bitstream operands

The bitstream (BS) datatype of Part 1 of this International Standard is assigned to the compressed colour specifier lists of tile array elements. These operands are compressed binary data objects. The Bitstream encoding method of this part is used to represent the bitstream datatype of Part 1. In general it is not possible to deduce the length of the compressed bitsteam from such information as the colour precisions or the number of cells in a tile. For this reason these operands are delimited in the same way that String operands are delimited (see 6.9.1), using SOS before the Bitstream operand and ST after.

6.16 Glyph mapping

Part 1 of this Standard specifies a list of code/glyphname pairs as one of the parameters of this element. The code/glyphname pairs in the list are encoded in this part with a run length format. Each code/glyphname pair in the list is encoded by: an octet (for the run count), an octet (for the code), and an integer (for the glyph name), all of which are encoded as integers in Basic format. If the run count is 1, then a single code/glyphname association is defined by the encoded pair. If the run count is greater than 1, then a sequence of code/glyphname associations is defined by the encoded pair. The base pair of the sequence is the encoded pair, and each of the two components of each pair in the sequence is 1 greater than the previous pair. The run count defines the number of pairs in the sequence, and is limited to 255 per sequence (for uniformity of results across encodings).

8 Representation of each element

Page 30

Clause 8, add the following to the notation used:

<symbol>(n)	-	exactly n occurrences, n=0,1,2,...

Page 30

Insert the following common grammar productions at the end of the page:

<line-measure-scalar>	=	<VDC: line aspect measure in VDC>
	|	{if LINE WIDTH SPECIFICATION MODE is absolute}
		<real: scale factor of device nominal line aspect measure>
	|	{if LINE WIDTH SPECIFICATION MODE is scaled}
		<real: line aspect measure as viewport fraction>
	|	{if LINE WIDTH SPECIFICATION MODE is fractional}
		<real: line aspect measure in mm>
	|	{if LINE WIDTH SPECIFICATION MODE is mm}
<marker-measure-scalar>	=	<VDC: marker aspect measure in VDC>
	|	{if MARKER SIZE SPECIFICATION MODE is absolute}
		<real: scale factor of device nominal marker aspect measure>
	|	{if MARKER SIZE SPECIFICATION MODE is scaled}
		<real: marker aspect measure as viewport fraction>
	|	{if MARKER SIZE SPECIFICATION MODE is fractional}
		<real: marker aspect measure in mm>
	|	{if MARKER SIZE SPECIFICATION MODE is mm}
<fill-measure-scalar>	=	<VDC: fill aspect measure in VDC>
	|	{if INTERIOR STYLE SPECIFICATION MODE is absolute}
		<real: scale factor of device nominal fill aspect measure>
	|	{if INTERIOR STYLE SPECIFICATION MODE is scaled}
		<real: fill aspect measure as viewport fraction>
	|	{if INTERIOR STYLE SPECIFICATION MODE is fractional}
		<real: fill aspect measure in mm>
	|	{if INTERIOR STYLE SPECIFICATION MODE is mm}
<edge-measure-scalar>	=	<VDC: edge aspect measure in VDC>
	|	{if EDGE WIDTH SPECIFICATION MODE is absolute}
		<real: scale factor of device nominal edge aspect measure>
	|	{if EDGE WIDTH SPECIFICATION MODE is scaled}
		<real: edge aspect measure as viewport fraction>
	|	{if EDGE WIDTH SPECIFICATION MODE is fractional}
		<real: edge aspect measure in mm>
	|	{if EDGE WIDTH SPECIFICATION MODE is mm}
<colour-specifier>	=	<integer: colour-index>
		{if COLOUR SELECTION MODE is indexed}
	|	<direct-colour-specifier>
		{if COLOUR SELECTION MODE is direct}

Representation of each element

<integer: colour-index>	=	<non-negative-integer>
<direct-colour-specifier>	=	<red green blue> {if COLOUR MODEL is RGB}
	|	<L A B> {if COLOUR MODEL is CIELAB}
	|	<L U V> {if COLOUR MODEL is CIELUV}
	|	<cyan magenta yellow black> {if COLOUR MODEL is CMYK}

Page 31

Subclause 8.1, add the following Delimiter element representations:

8.1.11 BEGIN COMPOUND PATH

<BEGIN-COMPOUND-PATH-opcode: 3/0 2/12>
<index: path-type>

<index: path-type>	=	<integer: 1> {text path}
	|	<integer: 2> {compound line}

8.1.12 END COMPOUND PATH

<END-COMPOUND-PATH-opcode: 3/0 2/13>

8.1.13 BEGIN PROTECTION REGION

<BEGIN-PROTECTION-REGION-opcode: 3/0 2/14>
<index: region-index>

<index: region-index> = <positive integer>

8.1.14 END PROTECTION REGION

<END-PROTECTION-REGION-opcode: 3/0 2/15>

8.1.15 BEGIN TILE ARRAY

<BEGIN-TILE-ARRAY-opcode: 3/0 03/0>
<point: position>
<enumerated: cell-path-direction>
<enumerated: line-progression-direction>
<integer: number-of-tiles-in-path-direction>
<integer: number-of-tiles-in-line-direction>
<integer: number-of-cells/tile-in-path-direction>
<integer: number-of-cells/tile-in-line-direction>
<real: cell-size-in-path-direction>
<real: cell-size-in-line-direction>
<integer: image-offset-in-path-direction>
<integer: image-offset-in-line-direction>
<integer: image-size-in-path-direction>
<integer: image-size-in-line-direction>

<enumerated: cell-path-direction> = <integer: 0> {0 degrees}
| <integer: 1> {90 degrees}
| <integer: 2> {180 degrees}
| <integer: 3> {270 degrees}

<enumerated: line-progression-direction> = <integer: 0> {90 degrees}
| <integer: 1> {270 degrees}

<integer: number-of-tiles-in-path-direction> = <positive integer>

<integer: number-of-tiles-in-line-direction> = <positive integer>

<integer: number-of-cells/tile-in-path-direction> = <positive integer>

<integer: number-of-cells/tile-in-line-direction> = <positive integer>

<real: cell-size-in-path-direction> = <positive real>

<real: cell-size-in-line-direction> = <positive real>

<integer: image-offset-in-path-direction> = <positive integer>

<integer: image-offset-in-line-direction> = <positive integer>

<integer: image-size-in-path-direction> = <non-negative integer>

<integer: image-size-in-line-direction> = <non-negative integer>

8.1.16 END TILE ARRAY

<END-TILE-ARRAY-opcode: 3/0 3/1>

Page 33

Subclause 8.2, COLOUR PRECISION, remove:

i.e., for each of the red, green and blue components.

Page 34

Sub-clause 8.2.10, change to read:

8.2.10 COLOUR VALUE EXTENT

<COLOUR-VALUE-EXTENT-opcode: 3/1 2/9>
<colour mapping specifier>

<colour mapping specifier> = <red green blue: minimum-colour-value>
<red green blue: maximum-colour-value>
{if COLOUR MODEL is 'RGB'}

<cyan magenta yellow black: minimum-colour-value>

Representation of each element

<cyan magenta yellow black: maximum-colour-value>
{if COLOUR MODEL is 'CMYK'}

<real: colour-scale-first-component>
<real: colour-offset-first-component>
<real: colour-scale-second-component>
<real: colour-offset-second-component>
<real: colour-scale-third-component>
<real: colour-offset-third-component>
{if COLOUR MODEL is 'CIELAB' or 'CIELUV'}

Page 33

Subclause 8.2: Add the following Metafile Descriptor element representations:

8.2.19 COLOUR MODEL

<COLOUR-MODEL-opcode: 3/1 3/3>
<index: colour-model>

<index: colour-model>	=	<integer: 1> {RGB}
	|	<integer: 2> {CIELAB}
	|	<integer: 3> {CIELUV}
	|	<integer: 4> {CMYK}

8.2.20 COLOUR CALIBRATION

<COLOUR-CALIBRATION-opcode: 3/1 3/4>
<real: reference-white-Xn>
<real: reference-white-Yn>
<real: reference-white-Zn>
<calibration-data>

<calibration-data>	=	<RGB-calibration-data> {if COLOUR MODEL indicator is RGB}
	|	<CMYK-calibration-data> {if COLOUR MODEL indicator is CMYK}
<RGB-calibration-data>	=	<RGB-calibration-matrix> <integer: n> {number-of-lookup-table-entries} <lookup-table-entry-R>(n) <lookup-table-entry-G>(n) <lookup-table-entry-B>(n)
<RGB-calibration-matrix>	=	<real: Xr> <real: Xg> <real: Xb> <real: Yr> <real: Yg> <real: Yb> <real: Zr> <real: Zg> <real: Zb>
<integer: n>	=	<non-negative integer>
<lookup-table-entry-R>	=	<colour-component: R'> <colour-component: R>
<lookup-table-entry-G>	=	<colour-component: G'> <colour-component: G>
<lookup-table-entry-B>	=	<colour-component: B'> <colour-component: B>

<CMYK-calibration-data>	=	<integer: n> {number of grid locations} <CMYK-grid-location>(n) <XYZ-grid-location>(n)
<CMYK-grid-location>(n)	=	<direct colour list>
<XYZ-grid-location>	=	<real: CIEXYZ-X> <real: CIEXYZ-Y> <real: CIEXYZ-Z>

8.2.21 FONT PROPERTIES

<FONT-PROPERTIES-opcode: 3/1 3/5>
<font-property-4-tuple>+

<font-property-4-tuple>	=	<index: property-indicator> <index: property-value-type> <property-value> <integer: priority>
<index: property-indicator>	=	<integer: 1> {font index}
	|	<integer: 2> {standard version}
	|	<integer: 3> {design source}
	|	<integer: 4> {font family}
	|	<integer: 5> {posture}
	|	<integer: 6> {weight}
	|	<integer: 7> {proportionate width}
	|	<integer: 8> {included glyph collections}
	|	<integer: 9> {included glyphs}
	|	<integer: 10> {design size}
	|	<integer: 11> {minimum size}
	|	<integer: 12> {maximum size}
	|	<integer: 13> {design group}
	|	<integer: 14> {structure}
<index: property-value-type>	=	<integer: 1> {integer}
	|	<integer: 2> {real}
	|	<integer: 3> {index}
	|	<integer: 4> {string}
	|	<integer: 5> {octet}
<property-value>	=	<index: font-index>
	|	<integer: standard-version>
	|	<string: design-source>
	|	<string: font-family>
	|	<index: posture>
	|	<index: weight>
	|	<index: proportionate-width>
	|	<included-glyph-collections>
	|	<included-glyphs>
	|	<real: design-size>
	|	<real: minimum-size>
	|	<real: maximum-size>
	|	<design-group>
	|	<index: structure>

Representation of each element

<index: font-index>	=	<positive integer>
<integer: standard-version>	=	<positive integer>
<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}
	|	<integer: >6> {reserved for registered values}
<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}
	|	<integer: >9> {reserved for registered values}
<index: proportionate-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}
	|	<integer: >9> {reserved for registered values}
<included-glyph-collections>	=	<integer: 1..(2^{32}-1)>+
<included-glyphs>	=	<integer: character-set-index>+
<integer: character-set-index>	=	<positive integer>
<real: design-size>	=	<positive real>
<real: minimum-size>	=	<positive real>
<real: maximum-size>	=	<positive real>
<design-group>	=	<octet: design-group-class>
		<octet: design-group-subclass>
		<octet: design-group-specific-group>
<octet: design-group-class>	=	<integer: 0..255>
<octet: design-group-subclass>	=	<integer: 0..255>
<octet: design-group-specific-group>	=	<integer: 0..255>

<index: structure> = <integer: 0> {undefined or not applicable}
| <integer: 1> {solid}
| <integer: 2> {outline}
| <integer: >2> {reserved for registered values}

8.2.22 GLYPH MAPPING

<GLYPH-MAPPING-opcode: 3/1 3/6>
<index: character-set-index>
<basis-set>
<integer: octets-per-code>
<code-glyph-name-run>+

<index: character-set-index> = <positive-integer>

<basis-set> = <enumerated: character-set-type>
{see 8.2.14, CHARACTER SET LIST}
<string: designation-sequence-tail>
{see 8.2.14, CHARACTER SET LIST}

<integer: octets-per-code> = <positive integer>

<code-glyph-name-run> = <run-count>
<character-code>
<afii-glyph-identifier>

<run-count> = <octet>

<character-code> = <octet>(octets-per-code)

<afii-glyph-identifier> = <integer: 1..(2^{32}-1)>

8.2.23 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 8.3.3, add the following to the end of the enumerated list for LINE WIDTH SPECIFICATION MODE:

| <integer: 2> {fractional}
| <integer: 3> {mm}

Page 37

Subclause 8.3.4, add the following to the end of the enumerated list for MARKER SIZE SPECIFICATION MODE:

| <integer: 2> {fractional}
| <integer: 3> {mm}

Page 37

Subclause 8.3.5, add the following to the end of the enumerated list for EDGE WIDTH SPECIFICATION MODE:

| <integer: 2> {fractional}
| <integer: 3> {mm}

Page 37

Subclause 8.3, add the following Picture Descriptor element representations:

8.3.16 INTERIOR STYLE SPECIFICATION MODE

<INTERIOR-STYLE-SPECIFICATION-MODE-opcode: 3/2 2/15>
<enumerated: specification-mode>

<enumerated: specification-mode> = <integer: 0> {absolute}
| <integer: 1> {scaled}
| <integer: 2> {fractional}
| <integer: 3> {mm}

Page 38

8.3.7, BACKGROUND COLOUR, replace:

<RGB: background colour>

with:

<direct-colour-specifier: background-colour> {see clause 8.0}

Page 38 (Am.1 Part 2 page 4)

8.3.11, LINE REPRESENTATION, delete the elaboration of <line-width-specifier> and change the '<line-width-specifier>' in the parameter list to:

<line-measure-scalar: line-width-specifier> {see clause 8.0}

Page 38 (Am.1 Part 2 page 4)

8.3.11, LINE REPRESENTATION, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 38 (Am.1 Part 2 page 4)

8.3.12, MARKER REPRESENTATION, delete the elaboration of <marker-size-specifier> and change the '<marker-size-specifier>' in the parameter list to:

<marker-measure-scalar: marker-size-specifier> {see clause 8.0}

Page 38 (Am.1 Part 2 page 4)

8.3.12, MARKER REPRESENTATION, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 38 (Am.1 Part 2 page 5)

8.3.13, TEXT REPRESENTATION, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 38 (Am.1 Part 2 page 5)

8.3.14, FILL REPRESENTATION, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 38 (Am.1 Part 2 page 6)

8.3.15, EDGE REPRESENTATION, delete the elaboration of <edge-width-specifier> and change the '<edge-width-specifier>' in the parameter list to:

<edge-measure-scalar: edge-width-specifier> {see clause 8.0}

Page 38 (Am.1 Part 2 page 6)

8.3.15, EDGE REPRESENTATION, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 39

Subclause 8.4, Add the following Control element representations:

8.4.13 PROTECTION REGION INDICATOR

<PROTECTION-REGION-INDICATOR-opcode: 3/3 03/0>

Representation of each element

<index: region-index>
<index: region-indicator>

<index: region-index> = <positive integer>

<index: region-indicator> = <integer: 0> {off}
| <integer: 1> {clip}
| <integer: 2> {shield}

8.4.14 GENERALIZED TEXT PATH MODE

<GENERALIZED-TEXT-PATH-MODE-opcode: 3/3 03/1>
<enumerated: text-path-mode>

<enumerated: text-path-mode> = <integer: 0> {off}
| <integer: 1> {non-tangential}
| <integer: 2> {axis-tangential}

8.4.15 MITRE LIMIT

<MITRE-LIMIT-OPCODE-opcode: 3/3 03/2>
<real: mitre-limit>

<real: mitre-limit> = <non-negative-real>

Page 40

8.4.3, AUXILIARY COLOUR, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see 8.0}

Page 41

Subclause 8.5, Add the following Graphical Primitive element representations:

8.5.22 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>

8.5.23 PARABOLIC ARC

<PARABOLIC-ARC-opcode: 3/4 2/11>
<point: tangent-intersection-point>
<point: start-point>

<point: end-point>

8.5.24 NON-UNIFORM B-SPLINE

<NON-UNIFORM-B-SPLINE-opcode: 3/4 2/12>
<integer: spline-order>
<integer: number-of-control-points>
<control-points>
<list-of-knots>
<real: parameter-start-value>
<real: parameter-end-value>

<integer: spline-order>	=	<positive integer> {m>0}
<integer: number-of-control-points>	=	<positive integer> {n≥m}
<control-points>	=	<point: control-point>(n)
<list-of-knots>	=	<real: knot>(m+n)

8.5.25 NON-UNIFORM RATIONAL B-SPLINE

<NON-UNIFORM-B-SPLINE-opcode: 3/4 2/13>
<integer: spline-order>
<integer: number-of-control-points>
<control-points>
<list-of-knots>
<real: parameter-start-value>
<real: parameter-end-value>
<weights>

<integer: spline-order>	=	<positive integer> {m>0}
<integer: number-of-control-points>	=	<positive integer> {n≥m}
<control-points>	=	<point: control-point>(n)
<list-of-knots>	=	<real: knot>(m+n)
<weights>	=	<real: weight>(n)

8.5.26 POLYBEZIER

<POLYBEZIER-opcode: 3/4 2/14>
<index: continuity indicator>
<control-points>

<index: continuity indicator>	=	<integer: 1> {discontinuous}
	|	<integer: 2> {continuous}
	|	<integer: 3> {smooth}
<control-points>	=	<point>(n) {n≥4}

8.5.27 SYMBOL

<SYMBOL-opcode: 3/4 2/15>

Representation of each element

<point: symbol-position>
<index: symbol-index>

<index: symbol-index> = <positive integer>

8.5.28 BITONAL TILE

<BITONAL-TILE-opcode: 3/4 3/0>
<integer: tile-identifier>
<index: compression-type>
<cell-background-colour>
<cell-foreground-colour>
<bitstream: tile>

<integer: tile-identifier> = <positive integer>

<index: compression-type> = <integer: 0> {null background}
| <integer: 1> {null foreground}
| <integer: 2> {T6}
| <integer: 3> {T4 1-dimensional}
| <integer: 4> {T4 2-dimensional}
| <integer: 5> {bitmap}
| <integer: 6> {LZW}
{>6, reserved for registered values}

<cell-background-colour> = <colour specifier> {see 8.4.3 AUXILIARY COLOUR}

<cell-foreground-colour> = <colour specifier> {see 8.4.3 AUXILIARY COLOUR}

8.5.29 TILE

<TILE-opcode: 3/4 3/1>
<integer: tile-identifier>
<index: compression-type>
<integer: cell-colour-precision>
<bitstream: tile>

<integer: tile-identifier> = <positive integer>

<integer: cell-colour-precision> = <non-negative integer>

<index: compression-type> = <integer: 0> {null background}
| <integer: 1> {null foreground}
| <integer: 2> {T6}
| <integer: 3> {T4 1-dimensional}
| <integer: 4> {T4 2-dimensional}
| <integer: 5> {bitmap}
| <integer: 6> {LZW}
{>6, reserved for registered values}

Page 42-43

Sub-clause 8.5.9, change:

<direct normal colour list> = <integer: 0>

<RGB: colour value>+

to:

<direct normal colour list> = <integer: 0>
<direct-colour-specifier: colour value>+

Page 42-43

Sub-clause 8.5.9, change:

<direct run> = <RGB: colour value>
<integer: number of repetitions>

to:

<direct run> = <direct-colour-specifier: colour value>
<integer: number of repetitions>

Page 46

Subclause 8.6, Add the following Primitive Attribute element representations:

8.6.37 LINE AND EDGE TYPE DEFINITION

<LINE-AND-EDGE-TYPE-DEFINITION-opcode: 3/5 2/8>
<index: line-type>
<real: dash-cycle-repeat-length>
<integer: list-of-dash-elements>

<index: line-type> = <negative-integer>

<integer: list-of-dash-elements> = <positive-integer>+

8.6.38 LINE CAP

<LINE-CAP-opcode: 3/6 3/15>
<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}
{>5, reserved for registered values}

<enumerated: dash-cap-indicator> = <integer: 0> {unspecified}
| <integer: 1> {match}

8.6.39 LINE JOIN

<LINE-JOIN-opcode: 3/5 2/10>

Representation of each element

<index: line-join-indicator>

<index: line-join-indicator>	=	<integer: 1> {unspecified}
	|	<integer: 2> {mitre}
	|	<integer: 3> {round}
	|	<integer: 4> {bevel}
		{>4, reserved for registered values}

8.6.40 LINE TYPE CONTINUATION

<LINE-TYPE-CONTINUATION-opcode: 3/5 2/11>
<index: continuation-mode>

<index: continuation-mode>	=	<integer: 1> {unspecified}
	|	<integer: 2> {continue}
	|	<integer: 3> {restart}
	|	<integer: 4> {adaptive continue}

8.6.41 LINE TYPE INITIAL OFFSET

<LINE-TYPE-INITIAL-OFFSET-opcode: 3/5 2/12>
<real: line-pattern-offset>

8.6.42 TEXT SCORE TYPE

<TEXT-SCORE-TYPE-opcode: 3/5 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}

8.6.43 RESTRICTED TEXT TYPE

<RESTRICTED-TEXT-TYPE-opcode: 3/5 2/14>
<index: restricted-text-type>

<index: restricted-text-type>	=	<integer: 1> {basic}
	|	<integer: 2> {boxed}
	|	<integer: 3> {isotropic}
	|	<integer: 4> {justified}

8.6.44 HATCH STYLE DEFINITION

<HATCH-STYLE-DEFINITION-opcode: 3/5 2/15>
<integer: hatch-index>
<enumerated: style-indicator>
<fill-measure-scalar: DX-first-vector>
<fill-measure-scalar: DY-first-vector>
<fill-measure-scalar: DX-second-vector>

<fill-measure-scalar: DY-second-vector>
<real: duty-cycle-length>
<integer: number-of-hatch-lines>
<list-of-gap-widths>
<list-of-line-types>

<integer: hatch-index> = <negative-integer>

<enumerated: style-indicator> = <integer: 0> {parallel}
| <integer: 1> {cross-hatch}

<integer: list-of-hatch-elements> = <positive-integer>+

<real: duty-cycle-length> = <positive real>

<integer: number-of-hatch-lines> = <positive-integer> {n}

<list-of-gap-widths> = <positive integer>(n)

<list-of-line-types> = <positive integer>(n)

8.6.45 GEOMETRIC PATTERN

<GEOMETRIC-PATTERN-opcode: 3/6 2/13>
<index: geometric-pattern-index>
<name: segment-identifier>
<pattern-extent>

<index: geometric-pattern-index> = <positive index>
<name: segment-identifier> = <integer>
<pattern-extent> = <point>(2)

8.6.46 INTERPOLATED INTERIOR DEFINITION

<INTERPOLATED-INTERIOR-DEFINITION-opcode: 3/6 2/14>
<index: style>
<reference-geometry-definition>
<integer: number-of-stages>
<stage-designator-list>
<reference-colour-list>

<index: style> = <integer: 1> {parallel}
| <integer: 2> {elliptical}
| <integer: 3> {triangular}
{>3, reserved for registered values}

<reference-geometry-definition> = <fill-measure-scalar>(2m) {$m \geq 0$}

<integer: number-of-stages> = <non-negative integer> {0 for triangular; number of stages otherwise}

<stage-designator-list> = <real>(m)

<reference-colour-list> = <colour-specifier>(k) {$k \geq 2$}

Representation of each element

8.6.47 EDGE CAP

<EDGE-CAP-opcode: 3/6 2/15>
<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}
		{>5, reserved for registered values}
<enumerated: dash-cap-indicator>	=	<integer: 0> {unspecified}
	|	<integer: 1> {match}

8.6.48 EDGE JOIN

<EDGE-JOIN-opcode: 3/6 3/3>
<index: edge-join-indicator>

<index: edge-join-indicator>	=	<integer: 1> {unspecified}
	|	<integer: 2> {miter}
	|	<integer: 3> {round}
	|	<integer: 4> {bevel}
		{>4, reserved for registered values}

8.6.49 EDGE TYPE CONTINUATION

<EDGE-TYPE-CONTINUATION-opcode: 3/6 3/4>
<index: continuation-mode>

<index: continuation-mode>	=	<integer: 1> {unspecified}
	|	<integer: 2> {continue}
	|	<integer: 3> {restart}
	|	<integer: 4> {adaptive continue}

8.6.50 EDGE TYPE INITIAL OFFSET

<EDGE-TYPE-INITIAL-OFFSET-opcode: 3/6 3/5>
<real: edge-pattern-offset>

8.6.51 SYMBOL LIBRARY INDEX

<SYMBOL-LIBRARY-INDEX-opcode: 3/6 3/6>
<index: symbol-library-index>

<index: symbol-library-index> = <positive integer>

8.6.52 SYMBOL COLOUR

<SYMBOL-COLOUR-opcode: 3/6 3/7>
<colour-specifier> {see 8.0}

8.6.53 SYMBOL SIZE

<SYMBOL-HEIGHT-opcode: 3/6 3/8>
<enumerated: symbol-scale-indicator>
<VDC: symbol-height>
<VDC: symbol-width>

<enumerated: symbol-scale-indicator> = <integer: 0> {height}
| <integer: 1> {width}
| <integer: 2> {both}

8.6.54 SYMBOL ORIENTATION

<SYMBOL-ORIENTATION-opcode: 3/6 3/9>
<VDC: x-component of up vector>
<VDC: y-component of up vector>
<VDC: x-component of base vector>
<VDC: y-component of base vector>

Page 55 (Am.1 Part 2 Page 9)

Subclause 8.9, add the following to the INHERITANCE FILTER LIST:

| <integer: 70> {line cap}
| <integer: 71> {line join}
| <integer: 72> {line type continuation}
| <integer: 73> {line type initial offset}
| <integer: 74> {text score type}
| <integer: 75> {restricted text type}
| <integer: 76> {generalized text path mode}
| <integer: 77> {interpolated interior definition}
| <integer: 78> {edge cap}
| <integer: 79> {edge join}
| <integer: 80> {edge type continuation}
| <integer: 81> {edge type initial offset}
| <integer: 82> {symbol library index}
| <integer: 83> {symbol colour}
| <integer: 84> {symbol size}
| <integer: 85> {symbol orientation}
| <integer: 86> {mitre limit}

Page 46

8.6.3, LINE WIDTH, delete the elaboration of <line-width-specifier> and change the '<line-width-specifier>' in the parameter list to:

<line-measure-scalar: line-width-specifier> {see clause 8.0}

Page 46

8.6.4, LINE COLOUR, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Representation of each element

Page 47

8.6.7, MARKER SIZE, delete the elaboration of <marker-size-specifier> and change the '<marker-size-specifier>' in the parameter list to:

<marker-measure-scalar: marker-size-specifier> {see clause 8.0}

Page 47

8.6.8, MARKER COLOUR, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 48

8.6.14, TEXT COLOUR, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 50

8.6.22, INTERIOR STYLE, add the enumerated values:

| <integer: 5> {geometric pattern}
| <integer: 6> {interpolated}

Page 50

8.6.23, FILL COLOUR, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 51

8.6.28, EDGE WIDTH, delete the elaboration of <edge-width-specifier> and change the '<edge-width-specifier>' in the parameter list to:

<edge-measure-scalar: edge-width-specifier> {see clause 8.0}

Page 51

8.6.29, EDGE COLOUR, delete the elaboration of <colour-specifier> and after the <colour-specifier> in the parameter list add:

{see clause 8.0}

Page 52

8.6.33, PATTERN SIZE, replace 'VDC:' in each of the parameter specifiers with 'fill-measure-scalar:'

Page 55 (Am.1 Part 2 Page 10)

Subclause 8.9, Add the following Segment Attribute element representation:

8.9.8 SEGMENT VISIBILITY

```
<SEGMENT-VISIBILITY-opcode: 3/8 2/12>
<name: segment-identifier>
<enumerated: segment-visibility>
```

<name: segment-identifier>	=	<integer>
<enumerated: segment-visibility>	=	<integer: 0> {visible} <integer: 1> {invisible}

Page 56

Clause 9, Defaults, modify COLOUR VALUE EXTENT for effect of other colour models:

COLOUR VALUE EXTENT:

if COLOUR MODEL is RGB,
minimum-colour-value 0,0,0
maximum-colour-value 63,63,63
if COLOUR MODEL is CMYK,
minimum-colour-value 0,0,0,0
maximum-colour-value 63,63,63,63
if COLOUR MODEL is CIELAB,
colour-scale-first-component 100.0/63
colour-offset-first-component 0.0
colour-scale-second-component ?
colour-offset-second-component ?
colour-scale-third-component ?
colour-offset-third-component ?
if COLOUR MODEL is CIELUV,
colour-scale-first-component 100.0/63
colour-offset-first-component 0.0
colour-scale-second-component ?
colour-offset-second-component ?
colour-scale-third-component ?
colour-offset-third-component ?

Page 59

Annex A, Formal grammar, add after <red green blue>:

<L A B>	::=	<bits for L^*> <bits for a^*> <bits for b^*>
<L U V>	::=	<bits for L^*> <bits for u^*>

Representation of each element

		<bits for v'>
<cyan magenta yellow black>	::=	<bits for cyan> <bits for magenta> <bits for yellow> <bits for black>

Page 59

Annex A, Formal grammar, delete duplicate colour definition by replacing:

<colour list> through <direct runlength bitstream colour list>

with:

<colour list> {see 8.5.9}

Page 60

Annex A, Formal grammar, change:

<RGB: colour value>

to

<direct-colour-specifier: colour value>

ISO/IEC 8632-3:1987/Am.3:1991

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer of Picture Description Information

Part 3

Binary Encoding

CONTENTS

5 Primitive data forms 1
6 Representation of abstract parameter types 2
7 Representation of each element 4

5 Primitive data forms

Page 17

Note 10, replace the second paragraph and the list with:

Real VDC and all real parameters of all elements may be coded with Fixed point reals (FXR) except the following, which must always be coded with floating point reals:

a) the *metric scaling factor* of the SCALE MODE element;

b) the *metric scale factor* of the DEVICE VIEWPORT SPECIFICATION MODE element.

Page 12

In section 5.3 change:

Each character is stored in an octet.

to:

Each character is stored in 1 or more consecutive octets, depending upon the coding of the particular character set. The following illustrates characters which are coded with 1 octet each.

6 Representation of abstract parameter types

Page 16

Replace the CD entry in Table 1 with the following 2 entries:

Abstract symbol	Parameter construction from primitive forms	Octets per parameter: symbol and value	Parameter range: symbol and value
CCO	UI at direct colour precision (dcp)	BCCO {=dcp/8}	CCOR {0..(2**dcp-1)} {see note 2}
CD	(CCO,CCO,CCO) or (CCO,CCO,CCO,CCO)	BCD ={3*BCCO} or BCD ={4*BCCO}	CCOR {see notes 1,16}

Page 16

Table 1, last line, change CDR to CCOR.

Page 16

Add the following at the end of table 1:

Abstract symbol	Parameter construction from primitive forms	Octets per parameter: symbol and value	Parameter range: symbol and value
BS	nUI at fixed precision (16-bit) {see note 15}	BBS {=2n}	BSR {see note 15}
OC	UI at fixed precision (8-bit)	BOC {=1}	OCR {0..255}
I32	UI at fixed precision (32-bit)	BI32 {=4}	I32R {0..2**32-1}

Page 16

Change note 2 in the Additional description (notes) for table 1 to:

2) For colour models RGB and CMYK a direct colour component is abstractly a real in the range {0,1}. For colour models CIELAB and CIELUV it is abstractly a real in the respective colour spaces with different ranges for the direct colour components. The COLOUR VALUE EXTENT element provides for the mapping between a direct colour component represented as UI and the corresponding real value.

Page 18

Add the following to the additional description (notes

Representation of abstract parameter types

15) The bitstream (BS) data type is encoded as a stream of binary digits (bits) packed in 16-bit unsigned integers. The BS data type is used in part 1 of this Standard for the compressed colour specifier lists of Tile Array elements. A bitstream type parameter shall be encoded in the Binary Encoding of this part with the smallest number of whole 16-bit words which will hold the bits of the parameter data. If the parameter data bits do not exactly fill an integral number of 16-bit words, the remaining bits in the last word shall be 0. The range for parameter type BS is not applicable.

16) The abstract parameter type CD is a 3-tuple or 4-tuple of CCO depending on COLOUR MODEL.

7 Representation of each element

Page 20

Subclause 7.2, add the following to Table 3, Delimiter Elements:

Element Class 0	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
BEGIN COMPOUND PATH	13	E	BE	{0,1}	see below
END COMPOUND PATH	14	n/a	0	n/a	n/a
BEGIN PROTECTION REGION	15	IX	BIX	IXR	see below
END PROTECTION REGION	16	n/a	0	n/a	n/a
BEGIN TILE ARRAY	17	P,	BP+	VDCR,	n/a
		2E,	2BE+	{0..3},{0,1},	n/a
		6I,2R,	6BI+2BR+	+IR,+RR,	n/a
		2I,2I	4BI	++IR,+IR	n/a
END TILE ARRAY	18	n/a	0	n/a	n/a

Code Description

13 BEGIN COMPOUND PATH: has 1 parameter:

P1: (index) path type: valid values are:

0 text path
1 compound line

See part 1, clause 6 of this Standard regarding defaults.

14 END COMPOUND PATH: has no parameters.

15 BEGIN PROTECTION REGION: has 1 parameter:

P1: (index) region index.

See part 1, clause 6 of this Standard regarding defaults.

16 END PROTECTION REGION: has no parameters.

17 BEGIN TILE ARRAY: has 13 parameters:

P1: (point) position.
P2: (enumerated) cell path direction.

0 0 degrees
1 90 degrees
2 180 degrees
3 270 degrees

P3: (enumerated) line progression direction.

0 90 degrees
1 270 degrees

P4: (integer) number of tiles in path direction.
P5: (integer) number of tiles in line direction.
P6: (integer) number of cells/tile in path direction.
P7: (integer) number of cells/tile in line direction.

Representation of each element

P8: (real) cell size in path direction.
P9: (real) cell size in line direction.
P10: (integer) image offset in path direction.
P11: (integer) image offset in line direction.
P12: (integer) image size in path direction.
P13: (integer) image size in line direction.

18 END TILE ARRAY: has no parameters.

Page 21

Replace the COLOUR VALUE EXTENT entry in Table 4, Metafile Descriptor Elements, with:

Element Class 1	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
COLOUR VALUE EXTENT	10	2CD or 6R	2BCD or 6BR	CCOR or RR	see below

Page 21

Subclause 7.3, add the following to Table 4, Metafile Descriptor Elements:

Element Class 1	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
COLOUR MODEL	19	IX	BIX	IXR	1
COLOUR CALIBRATION	20	3R,	3BR+	RR,	see below
		either			
		9R,I,6nCCO	9BR+BI+6nBCCO	RR,++IR,CCOR	
		or			
		I,nCD,3nR	BR+nBCD+3nBR	++IR,CCOR,RR	
FONT PROPERTIES	21	2IX,	2BIX+	+IXR	n/a
		n[S(or)IX(or)	n[BS(or)BIX(or)	SR(or)++IXR(or)	
		I32(or)nIX(or)	BI32(or)nBIX(or)	I32R(or)IXR(or)	
		R(or)3OC]	BR(or)3BOC]+	++RR(or)OCR	
		I	IR	++IR	
GLYPH MAPPING	22	IX,E,	BIX+BE+	+IXR,ER	n/a
		S,I,	BS+BI+	SR,+IR	
		n(OC,mOC,I32)	n((m+1)BOC+BI32)	OCR,OCR,I32R	
SYMBOL LIBRARY LIST	23	nS	nBS	SR	n/a

Page 22

Replace note 10, COLOUR VALUE EXTENT, **with:**

10 COLOUR VALUE EXTENT has variable parameters depending upon the colour model:

If the model is RGB or CMYK, then 2 parameters:

P1: (direct colour value) minimum colour value
P2: (direct colour value) maximum colour value

If the model is CIELAB or CIELUV, then 3 parameters:

P1: (real) scale and offset pair for first component.
P2: (real) scale and offset pair for second component.
P3: (real) scale and offset pair for third component.

See part 1, clause 6 of this Standard regarding defaults.

Page 22

Add the following notes (on Table 4):

Code Description

19 COLOUR MODEL: has 1 parameter:

P1: (index) colour model: the following values are standardized:

1 RGB
2 CIELAB
3 CIELUV
4 CMYK
>4 reserved for for registration and standardization.

20 COLOUR CALIBRATION: has 4 parameters

P1: (real) reference white value X component
P2: (real) reference white value Y component
P3: (real) reference white value Z component
P4: calibration data whose format and content depend upon the value of COLOUR MODEL. If the colour model is RGB then:

(real) 3x3 RGB calibration matrix: Xr, Xg, Xb, Yr, Yg, Yb, Zr, Zg, Zb.

(integer) number of lookup table entries (=n).

(colour component) 2n red lookup table entries: R', R.

(colour component) 2n green lookup table entries: G', G.

(colour component) 2n red lookup table entries: B', B.

If the colour model is CMYK then:

(integer) number of grid locations (=n).

(direct colour list) n CMYK grid locations.

(n*(3 real)) n XYZ grid locations, each being: CIEXYZ-X, CIEXYZ-Y, CIEXYZ-Z

See part 1, clause 6 of this Standard regarding defaults.

21 FONT PROPERTIES: has a variable number of parameter 4-tuples; each parameter 4-tuple contains:

P1: (index) property indicator, valid values are:

1 font index
2 standard version
3 design source
4 font family
5 posture
6 weight
7 proportionate width
8 included glyph collections
9 included glyphs

Representation of each element

10 design size
11 minimum size
12 maximum size
13 design group
14 structure
>14 reserved for registration and standardization

P2: (index) property value type, valid values are:

1 integer
2 real
3 index
4 string
5 octet
6 i32

P3: property value, following are property value type and property name in the correct association, followed by valid values where defined:

(index) font index

(integer) standard version, valid values are:

1 for ISO 9541:1991 (first version)

(string) design source

(string) font family

(index) posture, valid values are:

0 not applicable
1 upright
2 oblique
3 back slanted oblique
4 italic
5 back slanted italic
6 other
>6 reserved for registration and standardization

(index) weight, valid values are:

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
>9 reserved for registration and standardization

(index) proportionate-width, valid values are:

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
>9reserved for registration and standardization

(i32 list) included glyph collections: 1 or more AFII 32-bit glyph collection identifiers of type i32.

(index list) included glyphs: one or more character set indexes.

(real) design-size

(real) minimum-size

(real) maximum-size

(3 octets) design group: a 3-tuple of parameters of type octet, which respectively define the class, subclass, and specific group components of the design group.

(index) structure: valid values are:

0 undefined or not applicable
1 solid
2 outline
>2reserved for registration and standardization

P4: (integer) priority

21 GLYPH MAPPING: has 5 parameters:

P1: (index) character set index
P2: (enumerated) basis set character set type: valid values are as for CHARACTER SET LIST.
P3: (string) basis set designation sequence tail: valid values are as for CHARACTER SET LIST.
P4: (integer) octets per code, valid values are positive integers.
P5: (list of (octet,m*octet,i32)) code/glyphname pair list: a list which encodes pairs of code/glyphname associations. The form is a list of run length specifiers, where each specifier encodes a sequence of 1 or more code/glyphname pairs and is represented by: (octet,m*octet,i32). If the first octet equals 1, then only the single pair defined by the second and third parmeters is encoded (the second parameter is an m-octet code, where m is the value of P4). If the first octet is greater than 1, then the second and third parameters define the base pair of a sequence. Each of the two components of each pair in the sequence is 1 greater than the previous pair. The first octet is the number of pairs in the sequence.

23 SYMBOL LIBRARY LIST: has a variable parameter list

P1-Pn: n symbol library names (strings), the first name in the list is assigned to index 1, the second to index 2, etc.

Page 24

Table 5, last line, change CDR to CCOR.

Page 24

Table 5, LINE WIDTH SPECIFICATION MODE entry, change {0,1} to {0..3}.

Page 24

Table 5, MARKER SIZE SPECIFICATION MODE entry, change {0,1} to {0..3}.

Page 24

Table 5, EDGE WIDTH SPECIFICATION MODE entry, change {0,1} to {0..3}.

Representation of each element

Page 24

Add the following to the end of table 5, Picture Descriptor Elements:

Element Class 2	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
INTERIOR STYLE SPECIFICATION MODE	16	E	BE	{0,1,2,3}	0

Page 24

Note 3, LINE WIDTH SPECIFICATION MODE, add to the list of values for the parameter P1:

2 fractional
3 mm

Page 24

Note 4, MARKER SIZE SPECIFICATION MODE, add to the list of values for the parameter P1:

2 fractional
3 mm

Page 24

Note 5, EDGE WIDTH SPECIFICATION MODE, add to the list of values for the parameter P1:

2 fractional
3 mm

Page 24 (Am.1 page 4)

Note 11, LINE REPRESENTATION, replace the description of P3 with:

P1: (vdc) line width specifier if specification mode is 'absolute', (real) specifier otherwise.

Page 24 (Am.1 page 4)

Note 12, MARKER REPRESENTATION, replace the description of P3 with:

P1: (vdc) marker size specifier if specification mode is 'absolute', (real) specifier otherwise.

Page 24 (Am.1 page 5)

Note 15, EDGE REPRESENTATION, replace the description of P3 with:

P1: (vdc) edge width specifier if specification mode is 'absolute', (real) specifier otherwise.

Page 25

Note 7, BACKGROUND COLOUR, delete 'red,green,blue 3-tuple.'

Page 25

Add the following note (on Table 5):

Code Description

16 INTERIOR STYLE SPECIFICATION MODE: has 1 parameter:

P1: (enumerated) valid values are:

0 absolute
1 scaled
2 fractional
3 mm

Page 26

Add the following to table 6, control elements:

Element Class 3	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
PROTECTION REGION INDICATOR	17	2IX	2BIX	+IXR {0,1,2}	n/a,0
GENERALIZED TEXT PATH MODE	18	E	BE	{0,1,2}	0
MITRE LIMIT	19	R	BR	RR	1.00

Page 26

Note 3, AUXILIARY COLOUR, replace:

a 3-tuple of red, green, and blue integer (I) values

with:

a 3-tuple or 4-tuple of direct colour components (CCO)

Page 26

Add the following notes (on table 6):

Code Description

17 PROTECTION REGION INDICATOR: has 2 parameters:

P1: (index) region index
P2: (index) region indicator: valid values are:

0 off
1 clip
2 shield

18 GENERALIZED TEXT PATH MODE: has 1 parameter:

P1: (enumerated) text path mode: valid values are:

0 off
1 non-tangential
2 axis tangential

Representation of each element

19 MITRE LIMIT: has 1 parameter:

P1: (real) mitre limit

Page 28

Subclause 7.6, add the following to Table 7, Graphical Primitive Elements:

Element Class 4	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
HYPERBOLIC ARC	22	3P,4VDC	3BP+4BVDC	VDCR	n/a
PARABOLIC ARC	23	3P	3BP	VDCR	n/a
NON-UNIFORM B-SPLINE	24	2I,nP,	2BI+nBP+	IR,VDCR,	n/a
		(n+m)R	(n+m)BR+	++RR	n/a
		2R	2BR	++RR	n/a
NON-UNIFORM RATIONAL B-SPLINE	25	2I,nP,	2BI+nBP+	IR,VDCR,	n/a
		(n+m)R	(n+m)BR+	++RR	n/a
		2R	2BR+	++RR	n/a
		nR	nBR	++RR	n/a
POLYBEZIER	26	IX,nP	BIX+nBP	{1,2,3}, VDCR	n/a
SYMBOL	27	P,IX	BP+BIX	VDCR,+IXR	n/a
BITONAL TILE	28	I,IX	BI+BIX+	+IR,++IXR	n/a
		2CO	2BCO+	COR	n/a
		BS	BBS	BSR	n/a
TILE	29	I,IX	BI+BIX+	+IR,++IXR,	n/a
		I,BS	BI+BBS	+IR,BSR	n/a

Page 28

Add the following notes (on Table 7):

Code Description

22 HYPERBOLIC ARC: has 7 parameters:

P1: (point) centre point
P2: (point) transverse 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

23 PARABOLIC ARC: has 3 parameters:

P1: (point) tangent intersection point
P2: (point) start point
P3: (point) end point

24 NON-UNIFORM B-SPLINE: has a variable parameter list:

P1: (integer) spline order (=m)
P2: (integer) number of control points (=n)
P(3)-P(2+n): (points) array of control points
P(3+n)-P(2+2n+m): (real) list of knots, of length n+m.

P(3+2n+m): (real) parameter start value
P(4+2n+m): (real) parameter end value

25 NON-UNIFORM RATIONAL B-SPLINE: has a variable parameter list:

P1: (integer) spline order (=m)
P2: (integer) number of control points (=n)
P(3)-P(2+n): (points) array of control points
P(3+n)-P(2+2n+m): (real) list of knots, of length n+m.
P(3+2n+m): (real) parameter start value
P(4+2n+m): (real) parameter end value
P(5+2n+m)-P(4+3n+m): (real) list of weights, of length n.

26 POLYBEZIER: has a variable parameter list:

P1: (index) continuity indicator: valid values are:

1: discontinuous
2: continuous
3: smooth

P2-Pn: (point) list of point sequences: each sequence defines a single bezier curve and contains 4, 3, or 2 points respectively, according to the continuity indicator values 1, 2, 3.

27 SYMBOL: has 2 parameters:

P1: (point) position
P2: (index) symbol index

28 BITONAL TILE: has 5 parameters:

P1: (integer) tile identifier
P2: (index) compression type: valid values are:

0 null background
1 null foreground
2 T6
3 T4 1-dimensional
4 T4 2-dimensional
5 bitmap (uncompressed)
6 LZW
>6 reserved for registration and standardization

P3: (colour) cell background colour
P4: (colour) cell foreground colour
P5 (bitstream) compressed cell colour specifiers

29 TILE: has 4 parameters:

P1: (integer) tile identifier
P2: (index) compression type: valid values are:

0 null background
1 null foreground
2 T6
3 T4 1-dimensional
4 T4 2-dimensional
5 bitmap (uncompressed)
6 LZW
>6 reserved for registration and standardization

P3: (integer) cell colour precision: valid values are as for the local colour precision of CELL ARRAY.
P4: (bitstream) compressed cell colour specifiers

Representation of each element

Page 33

Table 8, COLOUR TABLE entry, change CDR to CCOR.

Page 33

Subclause 7.7, add the following to the end of Table 8, Primitive Attribute Elements:

Element Class 5	**Element Id**	**Parameter Type**	**Parameter List Length**	**Parameter Range**	**Default**
LINE AND EDGE TYPE DEFINITION	37	IX,R, nI	BIX+BR+ nBI	-IXR,RR ++IR	n/a
LINE CAP	38	IX,I	BIX+BE	+IXR,{0,1}	1,0
LINE JOIN	39	IX	BIX	+IXR	1
LINE TYPE CONTINUATION	40	IX	BIX	+IXR	1
LINE TYPE INITIAL OFFSET	41	R	BR	++RR	0.0
TEXT SCORE TYPE	42	nIX,nE	nBIX+nBE	IXR,{1..4}	1,1
RESTRICTED TEXT TYPE	43	IX	BIX	+IXR	1
HATCH STYLE DEFINITION	44	IX,E (4VDC or 4BR) R,I, nIX,nI	BIX+BE+ (4BVDC or 4R)+ BR+BI+ nBIX+nBI	-IXR,{0,1}, VDCR or RR, +RR,+IR, IXR,++IR	n/a
GEOMETRIC PATTERN	45	IX,N, 2P	BIX+BN+ 2BP	+IXR,NR VDCR	n/a
INTERPOLATED INTERIOR DEFINITION	46	IX,I, (2nVDC or 2nR) mR,kCO	2BIX+2BI+ (2nBVDC or 2nBR)+ mBR+kBCO	{1..3},+IR, VDCR or RR RR,COR	
EDGE CAP	47	IX,E	BIX+BE	+IXR,{0,1}	1,0
EDGE JOIN	48	IX	BIX	+IXR	1
EDGE TYPE CONTINUATION	49	IX	BIX	+IXR	1
EDGE TYPE INTIAL OFFSET	50	R	BR	++RR	0.00
SYMBOL LIBRARY INDEX	51	IX	BIX	+IXR	1
SYMBOL COLOUR	52	CO	BCO	COR	see below
SYMBOL SIZE	53	E,2VDC	BE+2BVDC	{0..2},VDCR	
SYMBOL ORIENTATION	54	4VDC	4BVDC	VDCR	0.1.1.0

Page 33

Note 3, LINE WIDTH, replace description of P1 with:

P1: (vdc) line width specifier if specification mode is 'absolute', (real) specifier otherwise; default as clause 6 part 1.

Page 33

Note 4, LINE COLOUR, replace:

a 3-tuple of red, green, and blue integer (I) values

with:

a 3-tuple or 4-tuple of direct colour components (CCO)

Page 34

Note 7, MARKER SIZE, replace description of P1 with:

P1: (vdc) marker size specifier if specification mode is 'absolute', (real) specifier otherwise; default as clause 6 part 1.

Page 34

Note 8, MARKER COLOUR, replace:

a 3-tuple of red, green, and blue integer (I) values

with:

a 3-tuple or 4-tuple of direct colour components (CCO)

Page 34

Note 14, TEXT COLOUR, replace:

a 3-tuple of red, green, and blue integer (I) values

with:

a 3-tuple or 4-tuple of direct colour components (CCO)

Page 35

Note 23, FILL COLOUR, replace:

a 3-tuple of red, green, and blue integer (I) values

with:

a 3-tuple or 4-tuple of direct colour components (CCO)

Page 36

Note 28, EDGE WIDTH, replace description of P1 with:

P1: (vdc) edge width specifier if specification mode is 'absolute', (real) specifier otherwise; default as clause 6 part 1.

Page 36

Note 29, EDGE COLOUR, replace:

a 3-tuple of red, green, and blue integer (I) values

with:

a 3-tuple or 4-tuple of direct colour components (CCO)

Representation of each element

Page 36

Note 33, in each of P1, P2, P3, P4 change:

(vdc)

to:

(vdc) or (real) specifier depending on the interior style specification mode, of

Page 37

Note 33, replace the last sentence with:

The default is as specified in clause 6 of part 1 of this Standard.

Page 37

Note 34, COLOUR TABLE, replace:

red,green,blue 3-tuples

with:

3-tuples or 4-tuples of direct colour components (CCO)

Page 37

Add the following notes (on Table 8):

Code Description

37 LINE AND EDGE TYPE DEFINITION: has a variable parameter list:

P1: (index) line type
P2: (real) dash cycle repeat length
P3-P(n+2): (integer) list of n dash elements

38 LINE CAP: has 2 parameters:

P1: (index) line cap indicator: the following values are standardized:

1 unspecified
2 butt
3 round
4 projecting square
5 triangle

P2: (enumerated) dash cap indicator: valid values are:

0 unspecified
1 match

39 LINE JOIN: has 1 parameter:

P1: (index) line join indicator: the following values are standardized:

1 unspecified
2 mitre

3 round
4 bevel

40 LINE TYPE CONTINUATION: has 1 parameter:

P1: (index) continuation mode: the following values are standardized:

1 unspecified
2 continue
3 restart
4 adaptive continue
>4 reserved for registration and standardization

41 LINE TYPE INITIAL OFFSET: has 1 parameter:

P1: (real) line pattern offset

42 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 right score
2 left score
3 through score
4 kendot

valid values for the score indicators are:

0 off
1 on

43 RESTRICTED TEXT TYPE: has 1 parameter:

P1: (index) restriction method: the following values are standardized:

1 basic
2 boxed
3 isotropic
4 justified

44 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: (vdc) first hatch direction vector x component if the interior style specification is 'absolute', or (real) x component otherwise; defaults are as specified in clause 6 of part 1 of this Standard.
P4: first hatch direction vector y component, specification units and defaults as for first hatch direction vector x component.
P5: second hatch direction vector x component, specification units and defaults as for first hatch direction vector x component.
P6: second hatch direction vector y component, specification units and defaults as for first hatch direction vector x component.
P7: (real) duty cycle length
P8: (integer) number of hatch lines (=n)
P9-P(8+n): (integers) list of n gap widths
P(9+n)-P(8+2n): (integers) list of n line types

45 GEOMETRIC PATTERN: has 4 parameters:

P1: (index) geometric pattern index
P2: (name) segment identifier

Representation of each element

P3: (point) first corner point
P4: (point) second corner point

46 INTERPOLATED INTERIOR DEFINITION: has a variable parameter list:

P1: (index) style: valid values are:

1 parallel
2 elliptical
3 triangular

P2: (integer) number of stages (=m)
P3: (2nVDC) reference geometry if interior style specification is 'absolute', or (2nR) otherwise; n=1 for parallel style and 2 for elliptical and triangular.
P4: (real) array of m stage designators
P5: (colour) array of k colour specifiers: k=3 for triangular, m+1 otherwise.

47 EDGE CAP: has 2 parameters:

P1: (index) edge cap indicator: the following values are standardized:

1 unspecified
2 butt
3 round
4 projected square cap
5 triangle

P2: (enumerated) dash cap indicator: valid values are:

0 unspecified
1 match

48 EDGE JOIN: has 1 parameter:

P1: (index) edge join indicator: the following values are standardized:

1 unspecified
2 mitre
3 round
4 bevel

49 EDGE TYPE CONTINUATION: has 1 parameter:

P1: (index) continuation mode: the following values are standardized:

1 unspecified
2 continue
3 restart
4 adaptive continue
>4 reserved for registration and standardization

50 EDGE TYPE INITIAL OFFSET: has 1 parameter:

P1: (real) pattern offset

51 SYMBOL LIBRARY INDEX: has 1 parameter:

P1: (index) symbol library index

52 SYMBOL COLOUR: has 1 parameter:

P1: (colour) symbol colour

53 SYMBOL SIZE: has 3 parameters:

P1: (enumerate) scale indicator: valid values are:

0 height
1 width

2 both

P2: (vdc) symbol height
P2: (vdc) symbol width

54 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

Page 39 (Am.1 page 7)

Section 7.10, add to the end of the INHERITANCE FILTER list:

70 line cap
71 line join
72 line type continuation
73 line type initial offset
74 text score type
75 restricted text type
76 generalized text path mode
77 interpolated interior definition
78 edge cap
79 edge join
80 edge type continuation
81 edge type initial offset
82 symbol library index
83 symbol colour
85 symbol size
86 symbol orientation
87 mitre limit

Page 39 (Am.1, page 7)

Subclause 7.10, add the following to Table 11, Segment Elements:

Element Class 8	Element Id	Parameter Type	Parameter List Length	Parameter Range	Default
SEGMENT VISIBILITY	8	N,E	BN+BE	NR,ER	-,0

Add the following notes (on Table 11):

Code Description

8 SEGMENT VISIBILITY: has 2 parameters:

P1: (name) segment identifier
P2: (enumerated) visibility: valid values are:

0 invisible
1 visible

Page 48

Add the following to the list of elements in annex C:

Representation of each element

Class	Element Code	Element Name
0	13	BEGIN COMPOUND PATH
0	14	END COMPOUND PATH
0	15	BEGIN PROTECTION REGION
0	16	END PROTECTION REGION
0	17	BEGIN TILE ARRAY
0	18	END TILE ARRAY
1	19	COLOUR MODEL
1	20	COLOUR CALIBRATION
1	21	FONT PROPERTIES
1	22	GLYPH MAPPING
1	23	SYMBOL LIBRARY LIST
2	16	INTERIOR STYLE SPECIFICATION MODE
3	17	PROTECTION REGION INDICATOR
3	18	GENERALIZED TEXT PATH MODE
3	19	MITRE LIMIT
4	22	HYPERBOLIC ARC
4	23	PARABOLIC ARC
4	24	NON-UNIFORM B-SPLINE
4	25	NON-UNIFORM RATIONAL B-SPLINE
4	26	POLYBEZIER
4	27	BITONAL TILE
4	28	TILE
4	29	SYMBOL
5	37	LINE AND EDGE TYPE DEFINITION
5	38	LINE CAP
5	39	LINE JOIN
5	40	LINE TYPE CONTINUATION
5	41	LINE TYPE INITIAL OFFSET
5	42	TEXT SCORE TYPE
5	43	RESTRICTED TEXT TYPE
5	44	HATCH STYLE DEFINITION
5	45	GEOMETRIC PATTERN
5	46	INTERPOLATED INTERIOR DEFINITION
5	47	EDGE CAP
5	48	EDGE JOIN
5	49	EDGE TYPE CONTINUATION
5	50	EDGE TYPE INTIAL OFFSET
5	51	SYMBOL LIBRARY INDEX
5	52	SYMBOL COLOUR
5	53	SYMBOL SIZE
5	54	SYMBOL ORIENTATION
8	8	SEGMENT VISIBILITY

ISO/IEC 8632-4:1987/Am.3:1991

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer of Picture Description Information

Part 4

Clear Text Encoding

CONTENTS

5 Metafile format 1
6 Encoding the CGM elements 5

5 Metafile format

Page 4

Clause 3, add to the notational conventions:

<...>(n) = exactly n occurrences, n=0,1,2,...

Page 8 (Am.1, page 1)

Subclause 5.3.1 after the first sentence add:

The data types OCTETS and I32 of part 1 of this Standard are bound to non-negative values of signed integers, also indicated in this encoding as I.

Page 9

Add new section 5.3.6:

5.3.6 Bitstream datatype

The parameters of type Bitstream, of tile array elements, shall be represented as follows. The bits taken 4 at a time are represented by a single hexidecimal digit in the Clear Text metafile. Null characters, SPACE, and format effector characters may be interspersed in the stream for readability. For example, a space character every 4 digits and a newline every 60 digits would provide well-formatted output. Bitstream datatype is indicated by "B" in the element definitions.

Page 10

Subclause 5.3.5, after <RED GREEN BLUE>, add:

<L A B> ::= <I:L><SEP><I:A><SEP><I:B>

<L U V> ::= <I:L><SEP><I:U><SEP><I:V>

<CYAN MAGENTA YELLOW BLACK> ::=<I:CYAN><SEP><I:MAGENTA>
<SEP><I:YELLOW><SEP><BLACK>

Page 10

Subclause 5.3.5, derived type for K, change:

K ::= <I> | <DIRECTCOLR> {colour value, depending on COLOUR SELECTION MODE}

<DIRECTCOLR> = <RED GREEN BLUE> {if COLOUR MODEL is RGB}
| <L A B> {if COLOUR MODEL is CIELAB}
| <L U V> {if COLOUR MODEL is CIELUV}

| <CYAN MAGENTA YELLOW BLACK> {if COLOUR MODEL is CMYK}

Page 11

Subclause 5.3.5: in the definition of 'V', change:

{ Used for line width, marker size, and edge width, ...

to:

{ Used for line width, marker size, fill interiors, and edge width, ...

Page 11

Subclause 5.4.1: Add the following words to the deleted words list:

CURVE
NORMALIZED
TANGENTIAL

Page 11

Subclause 5.4.3: Add the following words to the unabbreviated words list:

AXIS
BOTH
BEVEL
BEZIER
BITONAL
GLYPH
JOIN
LIMIT
MATCH
METHOD
MITRE
MODEL
OFFSET
POLYBEZIER
ROUND
SCORE
SHIELD
SPLINE
SYMBOL
TILE

Page 12

Subclause 5.4.4: Add the following abbreviations:

CALIBRATION	CALIB
COMPOUND	COMPO

Metafile format

CONTINUATION	CONT
DEFINITION	DEF
FRACTIONAL	FRAC
GENERALIZED	GEN
GEOMETRIC	GEO
HYPERBOLIC	HYPERB
INITIAL	INIT
INTERPOLATED	INTERP
LIBRARY	LIB
NON-UNIFORM B-SPLINE	NUB
NON-UNIFORM RATIONAL B-SPLINE	NURB
PARABOLIC	PARAB
PROPERTIES	PROP
PROTECTION	PROT
RATIONAL	RAT
UNIFORM	UNIF
UNSPECIFIED	UNSPEC

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 PROTECTION REGION	BEGPROTREGION	
END PROTECTION REGION	ENDPROTREGION	
BEGIN TILE ARRAY	BEGTILEARRAY	
END TILE ARRAY	ENDTILEARRAY	
COLOUR MODEL	COLRMODEL	
COLOUR CALIBRATION	COLRCALIB	
FONT PROPERTIES	FONTPROP	
GLYPH MAPPING	GLYPHMAP	
SYMBOL LIBRARY LIST	SYMBOLLIBLIST	
INTERIOR STYLE SPECIFICATION MODE	INTSTYLEMODE	
PROTECTION REGION INDICATOR	PROTREGION	
GENERALIZED TEXT PATH MODE	GENTEXTPATHMODE	
MITRE LIMIT	MITRELIMIT	
HYPERBOLIC ARC	HYPERBARC	
PARABOLIC ARC	PARABARC	
NON-UNIFORM B-SPLINE	NUB	
NON-UNIFORM RATIONAL B-SPLINE	NURB	
POLYBEZIER	POLYBEZIER	
SYMBOL	SYMBOL	
BITONAL TILE	BITONALTILE	
TILE	TILE	
LINE AND EDGE TYPE DEFINITION	LINEEDGETYPEDEF	
LINE CAP	LINECAP	
LINE JOIN	LINEJOIN	
LINE TYPE CONTINUATION	LINETYPECONT	
LINE TYPE INITIAL OFFSET	LINETYPEINITOFFSET	
TEXT SCORE TYPE	TEXTSCORETYPE	
RESTRICTED TEXT TYPE	RESTRTEXTTYPE	
HATCH STYLE DEFINITION	HATCHSTYLEDEF	
GEOMETRIC PATTERN	GEOPAT	
INTERPOLATED INTERIOR DEFINITION	INTERPINTDEF	
EDGE CAP	EDGECAP	
EDGE JOIN	EDGEJOIN	
EDGE TYPE CONTINUATION	EDGETYPECONT	
EDGE TYPE INITIAL OFFSET	EDGETYPEINITOFFSET	
SYMBOL LIBRARY INDEX	SYMBOLLIBINDEX	
SYMBOL COLOUR	SYMBOLCOLR	
SYMBOL SIZE	SYMBOLSIZE	
SYMBOL ORIENTATION	SYMBOLORI	
SEGMENT VISIBILITY	SEGVIS	

6 Encoding the CGM elements

Page 15

Subclause 6.2: Add the following Delimiter element encodings:

```
BEGIN COMPOUND PATH        ::=    BEGCOMPOPATH
                                  <SOFTSEP>
                                  <TEXT | LINE>
                                  <TERM>

END COMPOUND PATH          ::=    ENDCOMPOPATH<TERM>

BEGIN PROTECTION REGION    ::=    BEGPROTREGION
                                  <SOFTSEP>
                                  <I:REGIONINDEX>
                                  <SEP>
                                  <CLIP | SHIELD>
                                  <TERM>

END PROTECTION REGION      ::=    ENDPROTREGION<TERM>

BEGIN TILE ARRAY           ::=    <BEGTILEARRAY>
                                  <SOFTSEP>
                                  <P:POSITION>
                                  <SEP>
                                  <0 | 90 | 180 | 270>
                                  <SEP>
                                  <90 | 270>
                                  <SEP>
                                  <I:TILESPERPATH>
                                  <SEP>
                                  <I:TILESPERLINE>
                                  <SEP>
                                  <I:CELLSPERTILEPATH>
                                  <SEP>
                                  <I:CELLSPERTILELINE>
                                  <SEP>
                                  <R:CELLSPACING>
                                  <SEP>
                                  <R:LINESPACING>
                                  <SEP>
                                  <I:PATHOFFSET>
                                  <SEP>
                                  <I:LINEOFFSET>
                                  <SEP>
                                  <I:CELLSPERPATH>
                                  <SEP>
                                  <I:CELLSPERLINE>
                                  <TERM>

END TILE ARRAY             ::=    ENDTILEARRAY<TERM>
```

Page 15

Subclause 6.3: Add the following Metafile Descriptor element encodings:

```
COLOUR MODEL              ::=    COLRMODEL
                                 <SOFTSEP>
                                 <I:MODELINDEX>
                                  {1=RGB, 2=CIELAB,
                                   3=CIELUV, 4=CMYK,
                                   >4, reserved for registration}
                                 <TERM>

COLOUR CALIBRATION        ::=    COLRCALIB
                                 <SOFTSEP>
                                 <R:XN> <SEP> <R:YN> <SEP> <R:ZN>
                                 <SEP>
                                 <RGBCALIBDATA> | <CMYKCALIBDATA>
                                 <TERM>

<RGBCALIBDATA>            ::=    <RGBCALIBMATRIX>
                                 <SEP>
                                 < <I:LUTLENGTH> {n>0}
                                 <RLUTENTRY>(n)
                                 <GLUTENTRY>(n)
                                 <BLUTENTRY>(n) >
                          |      <I:LUTLENGTH> {n=0}

<RGBCALIBMATRIX>          ::=    <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>

<RLUTENTRY>               ::=    <SEP> <K:RPRIME> <SEP> <K:R>

<GLUTENTRY>               ::=    <SEP> <K:GPRIME> <SEP> <K:G>

<BLUTENTRY>               ::=    <SEP> <K:BPRIME> <SEP> <K:B>

<CMYKCALIBDATA>           ::=    <I:GRIDTABLELENGTH> {n}
                                 < <SEP> <DIRECTCOLR:CMYKGRIDLOCATION> >(n)
                                 <XYZGRIDLOCATION>(n)

<XYZGRIDLOCATION>         ::=    <SEP> <R:CIEXYZX> <SEP>
                                 <R:CIEXYZY> <SEP> <R:CIEXYZZ>

FONT PROPERTIES           ::=    FONTPROP
                                 <SOFTSEP>
                                 <PROPERTYRECORD>
                                 < <SEP> <PROPERTYRECORD> >*
                                 <TERM>

<PROPERTYRECORD>          ::=    <I:PROPERTY>
                                  {1=font index, 2=standard version,
                                   3=design source, 4=font family, 5=posture,
                                   6=weight, 7=proportionate width,
```

```
                                        8=included glyph collections,
                                        9=included glyphs, 10=design size,
                                        11=minimum size, 12=maximum size,
                                        13=design group, 14=structure,
                                        >14, reserved for registration}

                                       <SEP> <TYPEDVALUE> <SEP> <I:PRIORITY>

<TYPEDVALUE>               ::=         <<3:INDEX> <SEP> <I:FONTINDEX>>
                            |          <<1:INTEGER> <SEP> <I:STANDARDVERSION>>
                            |          <<4:STRING> <SEP> <S:DESIGNSOURCE>>
                            |          <<4:STRING> <SEP> <S:FONTFAMILY>>
                            |          <<3:INDEX> <SEP> <I:POSTURE>>
                                        {0=not applicable, 1=upright,
                                         2=oblique, 3=back slanted oblique,
                                         4=italic, 5=back slanted italic, 6=other,
                                         >6 reserved for registered values}

                            |          <<3:INDEX> <SEP> <I:WEIGHT>>
                                        {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,
                                         >9 reserved for registered values}

                            |          <<3:INDEX> <SEP> <I:PROPORTIONATEWIDTH>>
                                        {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,
                                         >9 reserved for registered values}

                            |          <<1:INTEGER> <SEP> <INCLUDGLYPHCOLLECT>>
                            |          <<3:INDEX> <SEP> <INCLUDGLYPHS>>
                            |          <<2:REAL> <SEP> <R:DESIGNSIZE>>
                            |          <<2:REAL> <SEP> <R:MINIMUMSIZE>>
                            |          <<2:REAL> <SEP> <R:MAXIMUMSIZE>>
                            |          <<3:INDEX> <SEP> <DESIGNGROUP>>
                            |          <<3:INDEX> <SEP> <I:STRUCTURE>>
                                        {0=undefined or not applicable,
                                         1=solid, 2=outline,
                                         >2 reserved for registered values}

<INCLUDGLYPHCOLLECT>       ::=         <I:GLYPH COLLECTION ID>
                                       <<SEP> <I:GLYPH COLLECTION ID>>*

<INCLUDGLYPHS>             ::=         <I:CHARSETINDEX>
                                       <<SEP> <I:CHARSETINDEX>>*

<DESIGNGROUP>              ::=         <I:CLASS> <SEP> <I:SUBCLASS>
                                       <SEP> <I:SPECIFICGROUP>

GLYPH MAPPING              ::=         GLYPHMAP
                                       <SOFTSEP>
```

<I:CHARSETINDEX>
<SEP>
<BASISSET>
<SEP>
<I:OCTETSPERCODE> {this defines n>0}
<<SEP> <CODENAMERUN>>+
<TERM>

<BASISSET> ::= <CHARSETDESIGNATOR>
{see CHARACTER SET LIST}

<CODENAMERUN> ::= <I:SEQUENCELENGTH> {1..255}
<SEP>
<I:OCTETCODE>
<SEP>
<I:AFII32BITIDENTIFIER>

NOTE — The code/glyphname pairs in the list are encoded in a runlength form, (N, CODE, NAME), N=1..255. If N=1, then a single code/glyphname association is defined by the encoded pair. If N>1, then a sequence of code/glyphname associations is defined by the encoded pair. The base pair of the sequence is the encoded pair, and each of the two components of each pair in the sequence is 1 greater than the previous pair. N is the number of pairs in the sequence, and is limited to 255 per sequence (for uniformity of results across encodings).

SYMBOL LIBRARY LIST ::= SYMBOLLIST
<OPTSEP>
<S:SYMBOLNAME>
<<SEP><S:SYMBOLNAME>>*
<TERM>

Page 16

Subclause 6.3, COLOUR PRECISION, change first two sentence of NOTE:

Colour direct values are 3*I or 4*I, depending on COLOUR MODEL. COLOUR PRECISION gives a single integer range, 0..MAXCOMPONENT, within which each of the three or four components is contained.

Page 17

Subclause 6.3, COLOUR VALUE EXTENT, replace:

COLOUR VALUE EXTENT ::= COLRVALUEEXT
<SOFTSEP>
<RGBCOLRMAP>
|
<LABCOLRMAP>
|
<LUVCOLRMAP>
|
<CMYKCOLRMAP>
<TERM>

<RGBCOLRMAP> ::= <RED GREEN BLUE:BLACK>
<SEP>
<RED GREEN BLUE:WHITE>

Encoding the CGM elements

<LABCOLRMAP>	::=	<R:COLRSCALE1> <SEP> <R:COLROFFSET1> <SEP> <R:COLRSCALE2> <SEP> <R:COLROFFSET2> <SEP> <R:COLRSCALE3> <SEP> <R:COLROFFSET3>
<LUVCOLRMAP>	::=	<LABCOLRMAP>
<CMYKCOLRMAP>	::=	<CYAN MAGENTA YELLOW BLACK:WHITE> <SEP> <CYAN MAGENTA YELLOW BLACK:BLACK>

Page 18

Subclause 6.4, BACKGROUND COLOUR, change:

<RED GREEN BLUE>

to

<DIRECTCOLR>

Page 18

Subclause 6.4, in LINE WIDTH SPECIFICATION MODE, MARKER SIZE SPECIFICATION MODE, EDGE WIDTH SPECIFICATION MODE, change:

`<ABSTRACT |SCALED>`

to:

`<ABSTRACT |SCALED |FRACTIONAL |MM>`

Page 18

Subclause 6.4: Add the following Picture Descriptor element encodings:

INTERIOR STYLE SPECIFICATION MODE	::=INTSTYLEMODE <SOFTSEP> <ABSTRACT |SCALED |FRACTIONAL |MM> <TERM>

Page 18

Subclause 6.5: Add the following Control element encodings:

PROTECTION REGION INDICATOR	::=	PROTREGION <SOFTSEP> <I:REGIONINDEX> <SEP> <I:REGIONINDICATOR>

		{0=off, 1=clip, 2=shield}
		<TERM>
GENERALIZED TEXT PATH MODE	::=	GENTEXTPATHMODE
		<SOFTSEP>
		<OFF | NONAXIS | AXIS>
		<TERM>
MITRE LIMIT	::=	MITRELIMIT
		<SOFTSEP>
		<R:MITRELIMIT>
		<TERM>

Page 19

Subclause 6.6: Add the following Graphical Primitive element encodings:

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	::=	NUB
		<SOFTSEP>
		<I:SPLINEORDER> {m}
		<I:NUMBERCONTROLPOINTS> {n≥m}
		<<SEP> <P:CONTROLPOINT>>(n)
		<<SEP> <R:KNOT>>(m+n)
		<SEP>
		<R:STARTVALUE>
		<SEP>
		<R:ENDVALUE>
		<TERM>

Encoding the CGM elements

NON-UNIFORM RATIONAL B-SPLINE ::=NURB
<SOFTSEP>
<I:SPLINEORDER> {m}
<I:NUMBERCONTROLPOINTS> {n≥m}
<<SEP> <P:CONTROLPOINT>>(n)
<<SEP> <R:KNOT>>(m+n)
<SEP>
<R:STARTVALUE>
<SEP>
<R:ENDVALUE>
<<SEP> <R:WEIGHT>>(n)
<TERM>

POLYBEZIER ::= POLYBEZIER
<SOFTSEP>
<I:CONTINUITYINDICATOR>
<<SEP> <P:CONTROLPOINT>>(n) {n≥1}
<TERM>

SYMBOL ::= SYMBOL
<SOFTSEP>
<P:POSITION>
<SEP>
<I:INDEX>
<TERM>

BITONAL TILE ::= BITONALTILE
<SOFTSEP>
<I:CMPRSNTYPE>
{0=null background, 1=null foreground,
2=T6, 3=T4 1-dimensional, 4=T4 2-dimensional,
5=bitmap, 6=LZW,
>6, reserved for registration}
<SEP>
<K:BACKGROUND>
<SEP>
<K:FOREGROUND>
<SEP>
<B:COMPRESSEDCELLS>
<TERM>

TILE ::= <TILE>
<SOFTSEP>
<I:CMPRSNTYPE>
{0=null background, 1=null foreground,
2=T6, 3=T4 1-dimensional, 4=T4 2-dimensional,
5=bitmap, 6=LZW,
>6, reserved for registration}
<SEP>
<I:CELLCOLRPREC>
<SEP>
<B:COMPRESSEDCELLS>
<TERM>

Subclause 6.7, COLOUR TABLE, change:

<RED GREEN BLUE>

to

<DIRECTCOLR>

Page 27

For PATTERN SIZE change both occurrences of <DELTAPAIR> to <V_DELTAPAIR> and after the element add the definition:

<V_DELTAPAIR>	::=	<OPTSEP> <V:DELTAX> <SEP> <V:DELTAY>

Page 24

Subclause 6.7: Add the following Primitive Attribute element encodings:

LINE AND EDGE TYPE DEFINITION	::=	LINEEDGETYPEDEF <SOFTSEP> <I:LINETYPE> <SEP> <R:REPEATLENGTH> <SEP> <I:DASHELEMENT> <<SEP> <I:DASHELEMENT>>* <TERM>
LINE CAP	::=	LINECAP <SOFTSEP> <I:CAP> {1=unspecified, 2=butt, 3=round, 4=projecting square, 5=triangle, >5 reserved for registration} <SEP> <UNSPECIFIED |MATCH> <TERM>
LINE JOIN	::=	LINEJOIN <SOFTSEP> <I:JOIN> {1=unspecified, 2=mitre, 3=round, 4=bevel, >4 reserved for registration} <TERM>
LINE TYPE CONTINUATION	::=	LINETYPECONT <SOFTSEP> <I:CONTMODE> {1=unspecified, 2=continue, 3=restart, 4=adaptive continue,

>4 reserved for registration}
<TERM>

LINE TYPE INITIAL OFFSET ::=

LINETYPEINITOFFSET
<SOFTSEP>
<R:PATTERNOFFSET>
<TERM>

TEXT SCORE TYPE ::=

TEXTSCORETYPE
<SOFTSEP>
< <I:SCORETYPE>
{1=right score, 2=left score,
3=through score, 4=kendot,
>4 reserved for registration}
<SEP>
<OFF |ON> >
<TERM>

RESTRICTED TEXT TYPE ::=

RESTRTEXTTYPE
<SOFTSEP>
<I:TEXTFIT>
{1=basic, 2=boxed,
3=isotropic, 4=justified,
>4 reserved for registration}
<TERM>

HATCH STYLE DEFINITION ::=

HATCHSTYLEDEF
<SOFTSEP>
<I:HATCHINDEX>
<SEP>
<PARALLEL |CROSSHATCH>
<SEP>
<V:FIRSTDIRX>
<SEP>
<V:FIRSTDIRY>
<SEP>
<V:SECONDDIRX>
<SEP>
<V:SECONDDIRY>
<SEP>
<R:DUTYCYCLE>
<SEP>
<I:NUMBEROFLINES> {this defines n}
<<SEP> <I:GAPWIDTH>>(n)
<<SEP> <I:LINETYPE>>(n)
<TERM>

GEOMETRIC PATTERN ::=

<GEOPAT>
<SOFTSEP>
<I:GEOPATINDEX>
<SEP>
<N:SEGID>
<SEP>
<P:FIRSTPOINT>
<SEP>
<P:SECONDPOINT>

		<TERM>
INTERPOLATED INTERIOR DEFINITION	::=	<INTERPINTDEF> <SOFTSEP> <I:INTSTYLE> <SEP> <<V:X><SEP><V:Y>>(1) {parallel} |<<V:X><SEP><V:Y>>(2) {elliptical, triangular} <SEP> <R:STAGEDESGN> (m) <SEP> <K:REFCOLRLIST> (m) <SEP> {m is the number of stages} <TERM>
EDGE CAP	::=	EDGECAP <SOFTSEP> <I:INDICATOR> {1=unspecified, 2=butt, 3=round, 4=projecting square, 5=triangle, >5 reserved for registration} <SEP> <UNSPECIFIED |MATCH> <TERM>
EDGE JOIN	::=	EDGEJOIN <SOFTSEP> <I:JOIN> {1=unspecified, 2=mitre, 3=round, 4=bevel, >4 reserved for registration} <TERM>
EDGE TYPE CONTINUATION	::=	EDGETYPECONT <SOFTSEP> <I:CONTMODE> {1=unspecified, 2=continue, 3=restart, 4=adaptive continue, >4 reserved for registration} <TERM>
EDGE TYPE INITIAL OFFSET	::=	EDGETYPEINITOFFSET <SOFTSEP> <R:PATTERNOFFSET> <TERM>
SYMBOL LIBRARY INDEX	::=	SYMBOLINDEX <SOFTSEP> <I:INDEX> <TERM>
SYMBOL COLOUR	::=	SYMBOLCOLR <SOFTSEP> <K:SYMBOLCOLOUR> <TERM>

Encoding the CGM elements

SYMBOL SIZE	::=	<SYMBOLSIZE> <SOFTSEP> <HEIGHT | WIDTH | BOTH> <SEP> <VDC:HEIGHT> <SEP> <VDC:WIDTH> <TERM>
SYMBOL ORIENTATION	::=	SYMBOLORI <SOFTSEP> <DELTAPAIR> {up vector} <SEP> <DELTAPAIR> {base vector} <TERM>

Page 24

Subclause 6.10: Add the following Segment Attribute element encodings:

SEGMENT VISIBILITY	::=	<SEGVIS> <SOFTSEP> <N:SEGID> <SEP> <VIS | INVIS> <TERM>

Page 30

Clause 7, Colour Value Extent default, change:

BLACK = 0,0,0
WHITE = 255,255,255

to

BLACK = 0,0,0
WHITE = 255,255,255, if COLOUR MODEL is RGB

WHITE = 0,0,0,0
BLACK = 1,1,1,1, if COLOUR MODEL is CMYK

APPENDIX 3

CGM AMENDMENT 4 ANSI PROJECT PROPOSAL

Project Proposal for Computer Graphics Metafiles (CGM) Amendment 4: Rules for Profiles

1 Identification of Proposed Project

1.1 Title:

CGM Amendment 4: Rules for Profiles.

1.2 Proposer:

X3H3 Technical Committee on Graphics Standards.

1.3 Date Submitted:

5 October 1991.

1.4 Project Type:

International, corresponding to project JTC1.24.5.21

2 Justification of Proposed Standard

2.1 Needs

Effective application of CGM requires the standard itself, ANSI/ISO 8632, in conjunction with a profile. A profile of CGM subsets of ISO 8632 and maximize the probability of successful interchange between CGM implementations. The current absence of rules for defining valid profiles of ISO 8632 has resulted in the existence of incorrect and inappropriate de-facto profiles of CGM. This in turn inhibits successful interoperability.

CGM profiles are also a necessary prerequisite for defining metafile generator and interpreter conformance requirements.

One prerequisite for a profile to become an ISP is that it be a valid and correct profile according to TR10000 and ISO procedures.. Amendment 4 is the normative addition to CGM for defining what comprises a valid correct profile of CGM.

2.2 Recommended Scope of the Standard

This normative annex to CGM (ANSI/ISO 8632) will include:

1) the rules for defining valid profiles of a CGM;

2) a Model Profile;

3) a sample Protocol Implementation Conformance Statement (PICS) Proforma;

4) criteria for evaluating CGM profiles;

5) generator and interpreter conformance statements to be added to ISO 8632 Part 1.

2.3 Existing Practice in Area of Proposed Standard

There is no existing practice in the area of rules for defining profiles of this functional standard.

2.4 Expected Stability of Proposed Standard With Respect to Current and Potential Technological Advances

The proposed rules for profiles (Amendment 4) will be built on ANSI/ISO 8632 (1987) as modified by Amendments 1 and 3 to produce ANSI/ISO 8632(1992) and TR10000.

3 Description of Proposed Project

3.1 Type of Document

Standard.

3.2 Definitions of Concepts and Special Terms

A CGM profile is a method for subsetting the specifications of a standard by identifying the elements, parameters, and options necessary for achieving a particular purpose.

3.3 Expected Relationship with Approved X3 Reference Models

This work will be consistent with the OSI Reference Model. It will also compliment the ISP process (ISO TR10000) and the ISO/IEC JTC1/SC24 Computer Graphics Reference Model (ISO DIS 11072).

3.4 Recommended Program of Work

Amendment 4 will be developed in conjunction with ISO/IEC JTC1/SC24 and in liaison with JTC1/SGFS. It will be processed according to ANSI procedures for the synchronous development of international standards.

3.5 Resources -- Individuals and Organizations Competent in Subject Matter

Members of the Virtual Device Interface Task Group (X3H3.3) of X3H3 are prepared to assist in the international development of this amendment. X3H3 is nominating a document editor of project JTC1.24.5.21. In addition, other members of X3H3 will be actively involved in the technical review of the working documents.

3.6 Recommended X3 Development Technical Committee

The international amendment will be developed by ISO/IEC JTC1/SC24/WG3. As U.S. Technical Advisory Committee to WG3, X3H3.3 will continue to provide U.S. contributions to this effort. Members of X3H3 will actively review and participate in the technical development.

3.7 Anticipated Frequency and Duration of Meetings

X3H3.3 meets in conjunction with X3H3 three times a year. The group has additional "ad hoc" meetings as necessary to meet milestones and timetables. In addition, representatives from X3H3.3 will attend ISO/IEC JTC1 SC24/WG3 meetings as called by SC24/WG3.

3.8 Target Date for CD to X3

This milestone will be met at the times that the ISO Committee Draft of Amendment 4 is circulated for vote and comment.

CD circulation: 10/91

3.9 Estimated Useful Life of Standard

This amendment will remain useful for the life of the associated functional standard, CGM -- at least 10 years.

4 Implementation Impacts

4.1 Impact on Existing User Practices and Investments

Amendment 4 will create a framework and mechanism for substantially improving CGM interoperability. There should be substantial positive impact on CGM implementations' interoperability due to the quality of CGM profiles which adhere to these rules.

4.2 Impact on Supplier Products and Support

This amendment should result in CGM products that can interoperate much more effectively. Because this amendment adds, for the first time, a conformance requirement on CGM generators and interpreters, suppliers will have to test their implementations for compliance with these conformance requirements.

4.3 Techniques and Costs for Compliance Verification

There is already technology available in the private sector and at NIST for CGM file testing which can be extended to support the new requirements contained in this amendment. Costs are expected to be similar to costs for existing Testing Services (e.g., Compiler Validation, GKS Validation).

4.4 Legal Considerations

There are no anticipated legal considerations.

5 Closely Related Standards Activity

5.1 Existing Standards

There are no existing rules for profiles for the CGM standard.

5.2 X3 Standards Development Projects

There are no X3 standards development projects within this scope other than those currently under development by X3H3.

5.3 X3/SPARC Study Groups

There are no related study groups.

5.4 Other Related Domestic Standards Efforts

There are no known related domestic standards efforts.

5.5 ISO Standards Development Projects

This work corresponds to project JTC1.24.5.21. The international rules for CGM profiles under development with U.S. participation will be adopted so that U.S. rules for CGM profiles will be identical to international profiles rules. No separately developed U.S. CGM profile rules are planned.

5.6 Other Related International Standards Development Projects

This work is one of a series of subprojects that address the standardization of the CGM, ISO project JTC1.24.5.

5.7 Recommendations for Coordinating Liaison

None.

5.8 Recommendations for Close Liaison

Close liaison will be carried out with X3V1 and X3L3 and the draft amendment produced under this project will be sent to X3V1 and X3L3 as part of the review process.

APPENDIX 4

CGM AMENDMENT 4 TEXT

Amendment 4 to ISO 8632-1

Information Processing Systems

Computer Graphics

Metafile for the Storage and Transfer

of Picture Description Information

Part 1

Functional Specification

CONTENTS

1 Scope 1

4 Concepts 2

5 Elements 3

7 Conformance 4

Annex D 6

Annex K 7

1 Scope

Page 4

add the following new paragraphs at the end:

Clause 7 of this International Standard defines conformance criteria for a Conforming Basic CGM Generator and a Conforming Basic CGM Interpreter.

Annex K of this Standard defines rules for profiles of CGM, and defines a Model Profile for graphical interchange.

4 Concepts

Page 40

Subclause 4.10, first line, insert before 'behaviour'

internal

Page 40

Subclause 4.10, first line, replace text after '—':

standardization of these aspects of generators and interpreters is beyond the scope of this standard, although this standard does define functional conformance requirements for conforming Basic generators and interpreters.

5 Elements

Page 43

Subclause 5.1, 5th pgph, insert after first sentence:

It also standardizes conformance criteria for functionally conforming generators and interpreters.

7 Conformance

Page 104

Add the following subclauses:

7.5 Conformance of metafile generators

Conformance of metafile generators is defined in terms of functional conformance to a profile of CGM.

Annex K defines rules for valid profiles of CGM. If "XXX" is a profile of CGM which conforms to the rules of Annex K, then a metafile generator is a Conforming "XXX" Generator if it:

a) generates metafiles which conform to the requirements of profile "XXX";

b) fully and accurately maps the graphical characteristics of application pictures onto a set of CGM elements which define those pictures.

With reference to this International Standard alone, there is only one defined notion of conforming CGM generator. Annex K of this standard contains a Model Profile of CGM. A metafile generator is a Conforming Basic Version n Generator (n=1,2,3) if it:

a) generates no syntax in violation of this International standard;

b) generates metafiles which conform to the version "n" subset of the Model Profile of Annex K of this standard;

c) fully and accurately maps the graphical characteristics of application pictures onto a set of CGM elements which define those pictures.

7.6 Conformance of metafile interpreters

Conformance of metafile interpreters is defined in terms of functional conformance to a profile of CGM.

Annex K defines rules for valid profiles of CGM. If "XXX" is a profile of CGM which conforms to the rules of Annex K, then a metafile interpreter is a Conforming "XXX" Interpreter if it:

a) is able to read any metafile which conforms to the requirements of profile "XXX";

b) fully and accurately renders the graphical characteristics of the CGM elements in any such metafile into a graphical image or picture.

With reference to this International Standard alone, there is only one defined notion of conforming CGM interpreters. Annex K of this standard contains a Model Profile of CGM. A metafile interpreter is a Conforming Basic Version n interpreter (n=1,2,3) if it:

a) is able to read any metafile which conforms to the version "n" subset of the Model Profile of Annex K of this standard;

b) fully and accurately renders the graphical characteristics of the CGM elements in any such metafile into a graphical image or picture.

Annex D

Page 123

Clause D.1, first pgph, replace second sentence with:

Although the functional behaviour of conforming generators and interpreters is standardized, the internal behaviour is not standardized, nor are fallback behaviours for either conforming interpreters dealing with invalid or incomplete metafile data, or interpreters which have insufficient functionality to deal with some valid metafile data.

Page 123

Clause D.1, first pgph, add at the end:

It is still considered valuable to present uniform suggestions for fallback strategies in such cases.

Annex K

Insert after Annex J:

ANNEX K

Rules for Profiles

(This annex is part of the Standard.)

TABLE OF CONTENTS

0 Introduction . . . . (pg # to be supplied)

- 0.1 Purpose . . . .
- 0.2 Objectives . . . .
- 0.3 Structure of this Annex . . . .

1 Scope and Field of Application . . . .

2 Normative References . . . .

3 Concept and Purpose of Profiles . . . .

4 Conformance . . . .

- 4.1 Conformance to this Annex . . . .
- 4.2 Conformance of metafiles to Profiles . . . .

5 Criteria for Evaluating Profiles . . . .

- 5.1 Criteria on the Profile in its Entirety . . . . . .
- 5.2 Criteria on the Technical Merit of the Profile . .

6 Structural Components of a Profile . . . .

- 6.1 Structure . . . .
- 6.2 Profile Content and Layout . . . .

7 Rules for Defining Profiles of ISO 8632 . . . .

- 7.1 General Principles for Defining Profiles . . . . .
- 7.2 Overall Specifications for Profiles . . . .
- 7.3 Element-by-element Specifications . . . .
- 7.4 Encoding-dependent Specifications. . . . .

8 Model Profile . . . .

0 Introduction

0.1 Purpose

This Annex to ISO 8632 provides rules for defining valid Profiles of ISO 8632. Profiles can be used as a method for subsetting ISO 8632 by identifying the CGM elements, parameters and options necessary for accomplishing a particular function.

0.2 Objectives

The primary objectives of this Annex 8632 are:

a) to promote interoperability by minimizing arbitrary subseting of ISO 8632;

b) to promote uniformity in the development of conformance tests;

c) to supplement the policies set forth in ISO/IEC/TR 10000 for International Standardized Profiles (ISPs);

d) to provide a basis for evaluating Profiles as potential ISPs.

0.3 Structure of this Annex

This Annex is composed of eight clauses which together provide the specifics for defining and evaluating Profiles of ISO 8632, including sections on:

a) the concept and purpose of a Profile;

b) the format and structure of a Profile;

c) rules and guidelines on the content of a Profile;

d) a Model Profile including a sample PICS Proforma;

e) criteria for evaluating Profiles.

1 Scope and Field of Application

This Annex to ISO 8632 defines the concept of Profiles to ISO 8632, provides rules for the structure and content of Profiles of ISO 8632, and defines criteria on which to evaluate Profiles of ISO 8632.

This Annex addresses the CGM data stream. It does not directly address the environmental or resource requirements of the CGM generator or CGM interpreter. These requirements should be specified in an Implementation Requirements Document, which may supplement the Profile.

Profiles of ISO 8632 may be defined by application constituencies solely interested in successful graphical interchange using ISO 8632. Alternatively, Profiles of ISO 8632 may be part of a set of inter-related standards and profiles assembled for the purpose of accomplishing a larger functional purpose. For example, an OSI Profile may specify a Profile of ISO 8632 as one of several OSI base standards in support of a specific requirement for interworking between systems.

This Annex does not define the application requirements or dictate application functional content of a Profile - the latter is the purview of application constituencies.

The Profile Rules (clause 7) and Model Profile (clause 8) of this Annex are limited in scope to defining rules for valid profiles for open interchange of graphical picture metafiles. Such application is the stated scope of ISO 8632. It is conceivable that other profiles could be defined for applying ISO 8632 for other purposes and goals, e.g., graphical object databases. However, such applications of ISO 8632 are beyond the scope of this Standard and the definition of such profiles is beyond the scope of this Annex.

CGM Generators and Interpreters shall be consistent with the Profile Rules and Model Profile specified herein. Generators shall produce conforming metafiles whose contents accurately represent the source picture according to the semantic rules of ISO 8632 and its Profile. Interpreters shall produce a target picture whose appearance agrees with the semantic rules for the meaning of the metafile.

This Annex to 8632 is intended for use by:

a) writers of profiles of 8632, including ISPs;

b) potential implementors of CGM generators and interpreters interested in improved interoperability and CGM interchange;

c) JTC1 - help to implement policies of TR10000.

2 Normative References

ISO/IEC/TR10000-1:1990 (E) Information Technology - *Framework and Taxonomy of International Standardized Profiles - Part 1: Framework.*

ISO/IEC/TR1000-2:1990 (E) Information Technology - *Framework and Taxonomy of International Standardized Profiles - Part 2: Taxonomy.*

ISO/IEC Directives Part 3:1989, *Drafting and Presentation of International Standards.*

3 Concept and Purpose of Profiles

A major goal of ISO 8632 is to facilitate the transfer of picture information between machines, sites, and applications. This goal may be impeded by the following:

a) ISO 8632 enables implementors to define subsets of the options of ISO 8632. Defining arbitrary subsets of ISO 8632 may produce different and incompatible dialects of CGM.

b) ISO 8632 enables implementors to specify private encodings, values and functions (e.g., private line types, escape elements, and GDP elements). Use of these types of private information inhibits interoperability.

c) ISO 8632 contains incomplete semantics and ambiguities. Interpretation of these semantics and ambiguities may differ among implementations and yield unpredictable results.

d) Implementations of ISO 8632 may generate a valid, but incompletely specified CGM (e.g., using color indicies and not including a color table). Interpretation of such a CGM file may yield unpredictable results.

Profiles provide a means to:

a) improve interoperability between implementations by inhibiting the proliferation of private subsets of ISO 8632;

b) provide a foundation for testing and promote uniformity of conformance tests for systems that implement the Profile;

c) enhance the availability, for procurement, of consistent implementations of Profiles.

A Profile of ISO 8632 defines the options, elements, and parameters of ISO 8632 necessary to accomplish a particular function and maximize the probability of interchange between systems implementing the Profile. Profiles are defined by application constituencies who agree to adhere to the same subset of CGM.

A Profile may:

a) give the meaning of *implementation dependent* semantics of some elements;

b) specify minima and maxima for lengths of elements that can have a variable amount of data;

c) specify subsets of the available elements that may be minimal and sufficient to accomplish a specific functional goal;

d) specify subsets or grouping of registered items from the ISO register.

A Profile of ISO 8632, according to the taxonomy of TR10000-2, is a Type F Profile, that is, an interchange format and representation Profile.

A Profile to ISO 8632 does not specify any requirement that would contradict or cause non-conformance to ISO 8632. Any metafile conforming to a Profile of ISO 8632 conforms to ISO 8632. In general, there will be files conforming to ISO 8632 that do not conform to Profiles of ISO 8632.

4 Conformance

4.1 Conformance to this Annex

In order to conform to this Annex, a Profile of ISO 8632 shall:

a) meet all requirements specified in this document;

b) be structured in accordance with the structural components and presentation rules defined in Clause 6 of this Annex;

c) not specify any requirements that would contradict or cause non-conformance to ISO 8632;

d) contain a conformance clause that may add requirements that are more specific and limited in scope that ISO 8632. Note:

A Profile may exclude valid optional capabilities and optional behavior permitted in ISO 8632;

e) meet the conformance requirements for a Type F Profile as defined in ISO/IEC/TR 10000-1;

f) meet the specific rules of Clause 7, herein, and be consistent with the Model Profile of Clause 8.

4.2 Conformance of metafiles to Profiles

In order to conform to a Profile of ISO 8632, a metafile shall:

a) contain all the mandatory elements specified in ISO 8632 and its Profile, and also any options of the Profile which it claims to include;

b) adhere to the conformance rules specified in ISO 8632 and its Profile.

5 Criteria for Evaluating Profiles

The uncontrolled proliferation of profiles can be nearly as damaging to widespread successful interoperability of ISO 8632 as the lack of a standard at all.

A major goal of this Annex, ISO/IEC TR10000, and ISO/IEC Directives, Part 3 is to facilitate and promote interoperability. Profiles are a means of achieving this goal.

The following criteria provide a means for determining the appropriateness and correctness of proposed Profiles. The objective is to limit the proliferation of profiles and ensure the quality of those profiles which will be implemented.

5.1 Criteria on the Profile in its Entirety

The following criteria are applied to the proposed Profile as a whole entity.

a) The application constituency and functional purpose of the proposed Profile must be well defined.

b) The functional purpose of the proposed Profile must not be satisfied by an existing Profile. If the functional purpose of a proposed Profile can be satisfied by a subset of an existing Profile, it should be so defined - significant subsets should not be replicated.

c) The proposed Profile must meet the identified functional requirements of the application constituency as so stated.

d) The proposed Profile must meet all requirements specified in this Annex.

5.2 Criteria on the Technical Merit of the Profile

The following criteria is applied to the technical content of the proposed Profile.

a) The proposed Profile shall not specify requirements that violate ISO 8632.

b) The proposed Profile shall place requirements on the CGM and not on implementations (e.g., generators and interpreters).

c) The specifications in the proposed Profile must be correct, complete, and well-defined.

d) The proposed Profile must be consistent in its requirements regarding CGM elements and parameters. For example, if the Profile places no restrictions on the number of indexes defined by the CHARACTER SET LIST element, then it is inconsistent to place a restriction on CHARACTER SET INDEX.

f) The proposed Profile must not specify requirements which are conflicting, unnecessary, or redundant.

g) The specification and restriction of CGM elements and parameters in the proposed Profile should follow common and accepted industry practice.

6 Structural Components of a Profile

6.1 Structure

A Profile of ISO 8632 shall contain the following components:

a) a concise definition of the scope of the function for which the Profile is defined, and of its purpose;

b) an illustration of the scenario within which the function is applicable; where a Profile is a member of a Group of Profiles, the scenario includes reference to the possibilities for interoperation that this provides;

c) normative reference to ISO 8632, including precise identification of the actual texts of ISO 8632 being used and of any approved amendments and technical corrigenda (errata), conformance to which is identified as potentially having an impact on achieving interoperation using the Profile;

d) informative reference to any other relevant source documents;

e) specifications of the application of ISO 8632, covering recommendations on the choice of classes or subsets and on the selection of options, ranges of parameter values, etc., and reference to registered objects. These specifications shall conform to rules of this Annex and shall be presented by reference to the Model Profile of Section 8 herein, that is, the statements of the Profile should endorse, accept, delete, replace, or modify those of the Model Profile;

f) a statement defining the requirements to be observed by systems claiming conformance to the Profile, including any remaining permitted options of ISO 8632, which thus become options of the Profile.

6.2 Profile Content and Layout

The content and layout of the Profile shall conform to the *Rules for Drafting and Presentation of International Standardized Profiles* in Annex A of ISO/IEC/TR 10000-1.

7 Rules for Defining Profiles of ISO 8632

The rules for defining valid Profiles of ISO 8632 are defined in the following sections of this clause.

7.1 General Principles for Defining Profiles

Compatibility with ISO 8632:
A Profile of ISO 8632 shall not contain any specifications which contradicts the normative specifications of ISO 8632.

Self-identifying:
Profiles shall require conforming metafiles to identify (e.g., within a Metafile Description string) that they adher to the Profile.

Source Identification:
Profiles shall require conforming metafiles to identify (e.g., within a Metafile Description string) the source of the metafile -- vendor, product, and product version.

Private encodings:
Private encodings shall not be specified by Profiles.

Defaults:
If a default is defined by ISO 8632, a Profile shall not change that default. If an element exists in ISO 8632 for setting the value of some parameter of the metafile or its environment, whose default in ISO 8632 is "implementation dependent", then that element shall be used if it is desired that a uniform default behavior be defined -- i.e., the generator shall either require the element be included in a Metafile Defaults Replacement or elsewhere in the metafile body -- the Profile shall not define a default if there is a mechanism for setting the value, but may require that the mechanism be used and may require a particular value. If there is no element to set a value (i.e., bundle tables in version 1 metafiles) then the Profile shall define the defaults.

Subsetting:
It is a principle role of Profiles of ISO 8632 to define subsets of the options of the Standard. In fact this is the only way that successful interoperability can be achieved, because some of the options of ISO 8632 include optional choices of element semantics and optional inclusion of private information. However subsetting shall not be arbitrary and shall have a clear connection to the achievement of one or more of the defined goals of the Profile. Subsetting of specifications which do not clearly lead to the achievement of the Profile goals shall be avoided.

Generator & Interpreter Behaviour:
Profiles of CGM shall address the elements of CGM, and not the resource requirements, performance or other behaviour characteristics of implementations of CGM.

Note -- it is nearly always true that statements regarding resource requirements of generators and interpreters -- e.g., the polyline point list maxima/minima which have been stated in the most prominent profiles of CGM to date -- can be stated as constraints on the elements themselves and not on the implementations.

Private Information:
A leading barrier to interoperability is the use within metafiles of the ISO 8632 options to include private information, both functions and parameter values. Profiles shall prohibit the use of any private information in conforming metafiles, unless the exact set of allowable

extensions and their precise definitions are defined within the profile.

Extensions:
Profiles shall not define extensions, as in the previous paragraph, as private elements or parameter values when the desired features are available as items in the Register of Graphical Items, or in the AFII Glyph Registry, the ISO 9541 Font Registry, or any other registry which is referenced in a normative manner by ISO 8632. In such cases the registry item shall be used. If there is another standard means within ISO 8362 to achieve the purposes of a prospective private item, and if this means meets the goals of the profile, then that means shall be used and not the private item. Example: private line types in version 3 metafiles, if the needed types could be defined by LINE & EDGE TYPE DEFINITION and associated elements.

Physical Media:
Physical file format and other issues of media, delivery, or networking are beyond the scope of ISO 8632 and shall not be specified by Profiles of ISO 8632. These issues are, however, critical for successful interoperability and should be addressed by the application community with supplemental specifications if necessary.

Subsetting Elements vs. Restricting Values:
In those cases where it is necessary to restrict an element to its default value in order to meet the goals of a Profile, the restriction should be achieved by allowing the element to appear in conforming files and restricting its value to the default rather than prohibiting the element. This applies especially to precision, type, and mode elements [Ed.N...next version: enumerate the CGM clause numbers of the affected elements] of the Metafile Descriptor, Picture Descriptor, and Control classes. For other elements profiles shall address whether the element is: prohibited; permitted; or required.

Note -- The implementation burden necessary to implement this guideline is small compared to the interoperability gain. Many implementations, even if only interested in the default value, consider such a "defensive" strategy to be good insurance against mistakes of other implementations in realizing the defaults of ISO 8632.

The rules for Profiles are all stated in terms of normative requirements that Profiles shall meet. [Ed.N. Would it be useful to also have guidelines, which could not by themselves cause non-conformance, but which would lead to "points" and "demerits" in rating the technical merit of a Profile? That is Profiles either fail (F), or pass with a grade (A, B, C)].

7.2 Overall Specifications for Profiles

This section contains specifications that pertain to valid Profiles of ISO 8632 but which do not fall conveniently under the category of a particular element in the Element-by-Element specifications of the following sections.

Mandatory for all Profiles:

Encodings:

Profiles of ISO 8632 shall address which of the standard encodings (8632/2, 8632/3, 8632/4) are permitted for metafiles conforming to the Profile.

[Model: all encodings].

Number of Pictures:

All Profiles shall address how many pictures are allowed in conforming metafiles (> 0, unlimited, exactly 1, >= 1 and <= n, >= 1 unlimited; and are empty metafiles, i.e., with zero pictures, permitted).

[Model: >= 1, unlimited].

Empty Pictures:

There are two kinds of empty pictures: 1) pictures with no graphical primitives; 2) pictures with no visible graphical primitives.

All Profiles shall address whether empty pictures, for each of these two kinds, and if permitted what their graphical effect is.

[Model: both kinds permitted; one blank frame of medium in background colour].

Optional:

Other things which profiles may optionally address: maximum size of metafiles; [...to be completed].

7.3 Element-by-element Specifications

Common Rules & Specifications:

The following common definitions and specifications are referenced by the individual element specifications which follow.

Zero-length geometric degeneracies in all line primitives: a line primitive element, whose entire locus resolves to a single point, denotes a graphical dot with diameter equal to the current line width and colour equal to the current line colour.

Zero-area geometric degeneracies in all filled area primitives: a filled area primitive element, whose entire locus resolves to a single point, denotes a graphical dot rendered/defined as follows... A filled area primitive element, whose entire locus resolves to a line (i.e., has zero area), [...to be completed per annex D]

Graphical Text Strings:

Rule:

length -- Profiles shall specify a limit for the maximum number of characters in graphical text strings, and shall specify the effect of null strings.

content -- Profiles shall conform to the Model Profile.

Model:

length -- strings shall contain at most 254 characters; null strings have no graphical effect.

content -- non-printing control codes shall be prohibited, except for NUL and the codes required to effect ISO 2022 character set switching consistent with the value of the CHARACTER CODING ANNOUNCER.

Non-graphical Text Strings: BegMet, BegPic, MetDesc, FontList,...

Rule:

length -- Profiles shall define the permitted number of characters.

content -- Profiles shall conform to the Model Profile.

Model:

length -- strings shall contain at most 1024 characters; [...to be completed].

content -- [...to be completed--CGM says the rules of character set selection do not apply to these strings, but some rules are needed if any information is to be passed in them].

Data Record StrinƋata.Escape & GDP (where allowed), Application

Rule:

length -- [...to be completed]

content -- Profiles shall conform to the Model Profile.

Model:

length -- strings shall contain at most 32767 characters; [..to be completed].

content -- [...to be completed].

Definitions for Individual Elements of ISO 8632:

[Editors note: eventually the "model" material in this section will be removed to clause 8 -- note that some general specification for the Model Profile already reside there]

[Editors note: there is much inconsistency in this first draft of the following specifications, and in fact the Model Profile probably violates the Profile Rules!].

ISO 8632 Clause Number	Element Name
5.2	**Delimiter Elements**
5.2.1	**BEGIN METAFILE**

Rule:

Usage of this element is mandated by ISO 8632.

Profiles shall follow specifications for non-graphical text strings in "Common Rules" regarding the Metafile Identifier parameter.

Model:

Per the specifications for non-graphical text strings in "Common Rules" regarding the Metafile Identifier parameter.

5.2.2 END METAFILE

Rule:

Usage of this element is mandated by ISO 8632. Profiles shall not address this element.

Model:

Nothing.

5.2.3 BEGIN PICTURE

Rule:

Usage of this element is mandated by ISO 8632 for non-trivial metafiles.

Profiles shall address the number of occurrences if the number of pictures is limited by the profile.

Profiles shall follow specifications for non-graphical text strings in "Common Rules" regarding Picture Metafile Identifier parameter.

Model:

There shall be at least one occurrence in conforming metafiles.

The maximum number of occurrences is not limited.

Per the specifications for non-graphical text strings in "Common Rules" regarding the Picture Identifier parameter.

5.2.4 BEGIN PICTURE BODY

Rule:

Profiles shall address the permissibility and effect of empty pictures.

Model:

Empty pictures are allowed and cause the occurrence of one image area of the graphical medium in the Background Colour.

5.2.5 END PICTURE

Rule:

Usage of this element is unambiguously defined by ISO 8632. Profiles shall not address this element.

Model:

Nothing.

====== Am.1 ======================================

5.2.6 BEGIN SEGMENT

Rule:

Profiles shall address whether this element is permitted, and if it is shall limit the maximum number of simultaneously defined segments (global and local included) at any point in the metafile.

Profiles may assign application meaning to the Segment Identifier parameter, but such meaning shall have no graphical effect.

Model:

Permitted.

At most 1024 segments may be simultaneously defined.

Segment Identifier has no meaning beyond being a unique identifier for the segment.

5.2.7 END SEGMENT

Rule:

Usage of this element is unambiguously defined by ISO 8632. Profiles shall not address this element.

Model:

Nothing.

5.2.8 BEGIN FIGURE

Rule:

Profiles shall address whether this element is permitted.

Profiles shall place limits on the number and identity of elements comprising a figure definition, and shall consider consistency with other filled area primitives (e.g., POLYGON) in doing so.

Model:

Permitted.

At most [...what is a reasonable number?] elements, no restriction on which eligible ISO 8632 elements may be included.

5.2.9 END FIGURE

Rule:

Usage of this element is unambiguously defined by ISO 8632. Profiles shall not address this element.

Model:

Nothing.

======= Am.3 ==

5.2.10 BEGIN COMPOUND PATH

Rule:

Profiles shall place limits on the number and identity of elements comprising a path definition, and shall consider consistency with other line primitives in doing so.

Model:

Permitted.

At most [...what is a reasonable number?] elements, no restriction on which eligible ISO 8632 elements may be included.

5.2.11 END COMPOUND PATH

Rule:

Usage of this element is unambiguously defined by ISO 8632. Profiles shall not address this element.

Model:

Nothing.

5.2.12 BEGIN PROTECTION REGION

Rule:

Profiles shall address whether this element is permitted, and if it is shall limit the maximum number of simultaneously defined protection regions.

Profiles may assign application meaning to the Region Identifier parameter, but such meaning shall have no graphical effect.

Model:

Permitted.

At most 32 regions may be simultaneously defined.

Region Identifier has no meaning beyond unique identifier.

5.2.13 END PROTECTION REGION

Rule:

Usage of this element is unambiguously defined by ISO 8632. Profiles shall not address this element.

Model:

Nothing.

5.2.14 BEGIN TILE ARRAY

Rule:

Profiles shall address whether tile arrays are permitted/prohibited.

If permitted, profiles shall place restrictions on:

Number of cells in path and line direction;
Number of tiles in path and line direction.

If permitted, profiles may place restrictions on:
Path and line progression directions;
Image offset parameters.

Model:

Permitted.
No limits on pel path and line progression directions.
Number of pels per line limited to [...what is reasonable?]
Number of lines limited to [...what is reasonable?]
Number of tiles path direction limited to [...what is reasonable?]
Number of tiles line direction limited to [...what is reasonable?]
No restrictions on image offset.

5.2.15 END TILE ARRAY

Rule:

Usage of this element is unambiguously defined by ISO 8632.
Profiles shall not address this element.

Model:

Nothing.

5.3 Metafile Descriptor Elements

5.3.1 METAFILE VERSION

Rule:

Profiles shall follow Model Profile on Number of occurrences.

Profiles may restrict valid values of version to exactly one of the values 1, 2, 3. The element content, state requirements, and parameter values of conforming files shall then be consistent.

Model:

This element shall appear exactly once.

Any of the values 1, 2, 3 may be used.

5.3.2 METAFILE DESCRIPTION

Rule:

Profiles shall follow Model Profile on Number of occurrences and information content (however each Profile shall define it's own identifying substring and substitute it for the Model Profile's).

Profiles shall follow specifications for non-graphical text strings in "Common Rules" regarding Picture Metafile Identifier parameter.

Model:

There shall be exactly one occurrence of this element.

There shall be a recognizable sub-string "Model-Profile-1" (case insensitive) for all files conforming to this Model Profile. Other profiles shall define an identifier string and require its presence conforming metafiles.

There shall be a recognizable sub-string which uniquely identifies the metafile source (minimally vendor, product, version, and optionally other useful information such as date, user, etc).

String content shall follow the specifications for non-graphical text strings in "Common Rules".

5.3.3 VDC TYPE

Rule:

Element shall be permitted by all Profiles.

Profiles may restrict parameter value if there is cause to do so.

Model:

Element is permitted.

No restriction on parameter value.

5.3.4 INTEGER PRECISION

Rule:

Permissibility -- per the Model Profile.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

Element is permitted.

[Binary: Values are restricted to 16 and 32]. [Character: ...] [Clear Text: ...]

5.3.5 REAL PRECISION

Rule:

Permissibility -- per the Model Profile.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

[Binary: Value is restricted to (1,16,16) and (0,9,23).] [Character: ...] [Clear Text: ...]

5.3.6 INDEX PRECISION

Rule:

Element shall be permitted by all Profiles.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

[Binary: Values are restricted to 8, 16 and 32]. [Character: ...] [Clear Text: ...]

5.3.7 COLOUR PRECISION

Rule:

Element shall be permitted by all Profiles.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

[Binary: Values are restricted to 8, 16]. [Character: ...] [Clear Text: ...]

5.3.8 COLOUR INDEX PRECISION

Rule:

Element shall be permitted by all Profiles.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

[Binary: Values are restricted to 8, 16]. [Character: ...] [Clear Text: ...]

5.3.9 MAXIMUM COLOUR INDEX

Rule:

Element shall be permitted by all Profiles.

Profiles shall place limit on parameter value.

Note -- the limit required by the profile should be consistent with COLOUR INDEX PRECISION (...or is "field width" an orthogonal issue?) and COLOUR TABLE specifications.

Model:

Element is permitted.

Values are restricted to (0,1) for monochrome metafiles, (0-63) for gray scale metafiles, and (0-255) for colour.

5.3.10 COLOUR VALUE EXTENT

Rule:

Element shall be permitted by all Profiles.

Profiles may constrain parameter values if there is cause to do so.

Model:

Element is permitted.

Values may be any valid ISO 8632 values which are consistent with the COLOUR PRECISION.

5.3.11 METAFILE ELEMENT LIST

Rule:

Element is mandated by ISO 8632.

Profiles may require element to be exact.

Model:

No contraints.

5.3.12 METAFILE DEFAULTS REPLACEMENT

Rule:

Profiles may require each occurrence of MDR to define just one default.

Profiles may restrict the syntax in the encodings.

Note -- profiles should consider whether the setting of values in the MDR is consistent with other specifications, such as the number of pictures allowed in the metafile.

Model:

[Binary: the element shall not be partitioned, nor shall any of its contained elements].

5.3.13 FONT LIST

Rule:

Element shall be required by all Profiles.

Profiles shall conform to the model profile regarding the rules of usage and syntax of font names.

Profiles shall define a specific maximum limit for the number of fonts in the font list.

Profiles shall specify a specific set of fonts which are permitted for use in the font list.

Model:

Element shall be present in conforming metafiles.

Rules of usage -- All fonts referenced in the metafile, including the default, (nominally index 1) shall be defined.

Font name syntax -- Font name construction shall be consistent with the rules of ISO 9541.

The maximum number of fonts in the font list shall be 64. The fonts which may be used in the font list shall be selected from the set: [...to be specified by each Profile...]

5.3.14 **CHARACTER SET LIST**

Rule:

Element shall be required by all Profiles.

Profiles shall conform to the model profile regarding the rules of usage.

Profiles shall conform to the model profile regarding acceptable character set types.

Profiles shall define a specific maximum limit for the number of character sets in the character set list.

Profiles shall specify a specific set of character sets which are permitted for use in the character set list. Acceptable Character Set Types -- "complete codes" shall be prohibited by Profiles unless the content of the complete code, and its associated designation sequence tail, is clearly defined within the Profile.

Model:

Element shall be present in conforming metafiles.

Rules of usage -- All character sets referenced in the metafile, including the default, (nominally index 1) shall be defined (1 is defined to have a default by ISO 8632, and this default is consistent with ISO 646 IRV and is acceptable if the national code positions are not referenced).

The maximum number of character sets in the list shall be [...to be specified by each Profile...].

The character sets which may be used in the list shall be selected from the set: ISO Registry of Character Sets [extensible by Profiles].

5.3.15 CHARACTER CODING ANNOUNCER

Rule:

Element shall be permitted by all Profiles.

Profiles may constrain parameter values if there is cause to do so.

Profiles shall prohibit private values unless the precise meaning of the private value is clearly defined in the profile.

Model:

====== Am.1 ======

5.3.16 NAME PRECISION

Rule:

Element shall be permitted by all Profiles.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

Element is permitted.

[Binary: ...] [Character: ...] [Clear Text: ...]

5.3.17 MAXIMUM VDC EXTENT

Rule:

Profile may restrict values.

Model:

[...to be completed...]

5.3.18 SEGMENT PRIORITY EXTENT

Rule:

Profile may restrict values.

Model:

[...to be completed...]

======= **Am.3** ==

5.3.19 COLOUR MODEL

Rule:

Profiles shall address whether element is prohibited/permitted. Profiles may restrict the set of colour models if there is cause to do so.

Model:

Permitted; unrestricted.

5.3.20 COLOUR CALIBRATION

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted, profiles shall define the maximum length of calibration arrays.

Model:

Permitted; arrays <= 256 points.

5.3.21 FONT PROPERTIES

Rule:

Profiles shall address whether element is prohibited/permitted. Profiles shall define the legal subset of parameter values if all are not to be supported by the profile.

Model:

Permitted; all defined index and enumerated values of all parameters shall be supported.

5.3.22 GLYPH MAPPING

Rule:

Profiles shall address whether element is prohibited/permitted. Profiles shall specify what subset of the AFII registered glyphs may be referenced in conforming metafiles. Profiles shall define a maximum limit on the number of glyphs which may be defined.

Model:

Permitted; any AFII-registered glyphs; at most 8192 glyphs may be invoked at once.

5.3.23 SYMBOL LIBRARY LIST

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted, Profiles shall define what libraries may be accessed and shall define unambiguous naming procedures for those libraries. Profiles shall define the maximum number of libraries which may be accessed simultaneously.

Model:

Permitted; naming of libraries shall be consistent with naming procedures defined in ISO 9541. The set of libraries which may be referenced is [...to be completed by the real Profiles].

5.4 Picture Descriptor Elements

5.4.1 SCALING MODE

Rule:

Profiles shall address whether element is prohibited/permitted/required.

Model:

Permitted.

5.4.2 COLOUR SELECTION MODE

Rule:

Permissibility -- Profiles shall follow the Model. Values -- Profiles shall allow all values unless this specifically violates a goal of the Profile. State -- Profiles may restrict the changing of colour mode and the states in which it may be defined.

Model:

Permitted; all values allowed; at most one mode per picture, defined in the Picture Descriptor; modes of referenced global segments shall be consistent.

5.4.3 LINE WIDTH SPECIFICATION MODE

Rule:

Permissibility -- Profiles shall follow the Model. Values -- profiles shall allow all values unless this specifically violates a goal of the profile. State -- Profiles shall not restrict the changing of mode or position of occurrence of the element unless allowing such violates another goal of the Profile.

Model:

Permitted; all values allowed; no state or position restrictions.

5.4.4 MARKER SIZE SPECIFICATION MODE

[to be defined consistently with LINE WIDTH SPECIFICATION MODE]

5.4.5 EDGE WIDTH SPECIFICATION MODE

[to be defined consistently with LINE WIDTH SPECIFICATION MODE]

5.4.6 VDC EXTENT

Rule:

Permissibility -- Profiles shall follow the Model. Values -- Profiles shall address whether reflections of coordinate space and placement of origin other than lower left are permitted.

Model:

Permitted; no restrictions.

5.4.7 BACKGROUND COLOUR

Rule:

Permissibility -- per Model Profile. Values -- per Model Profile.

Model:

Required or prohibited, consistent with each metafile's choice of the "all or none" colour definition rule. Values shall be consistent with each metafile's choice of monochrome, grayscale, full colour.

======= Am.1 ===

5.4.8 DEVICE VIEWPORT

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted, Profiles shall define interaction of the element with environmental presentation directives and meaning if specified value is inconsistent with presentation device.

Model:

Prohibited.

Note -- while it is a general principle that the model profile does not restrict elements or options, this element is highly device dependent, potentially conflicts with directives of client applications, and is the subject of debate concerning whether the element should remain in ISO 8632.

5.4.9 DEVICE VIEWPORT SPECIFICATION MODE

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted, Profiles shall define the set of legal values.

Model:

Prohibited.

5.4.10 DEVICE VIEWPORT MAPPING

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted, Profiles shall define the set of legal values.

Model:

Prohibited.

5.4.11 LINE REPRESENTATION

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted Profiles shall define the maximum number of simultaneous bundle definitions.

If prohibited, Profiles shall define the defaults for indexes 1..5. Complete Definition -- per model profile.

Model:

Permitted; <= 20 definitions. Complete Definition -- any referenced bundle index must be defined.

5.4.12 MARKER REPRESENTATION

[to be defined consistently with LINE REPRESENTATION]

5.4.13 TEXT REPRESENTATION

[to be defined consistently with LINE REPRESENTATION]

5.4.14 FILL REPRESENTATION

[to be defined consistently with LINE REPRESENTATION]

5.4.15 EDGE REPRESENTATION

[to be defined consistently with LINE REPRESENTATION]

====== Am.3 ======

5.4.16 INTERIOR STYLE SPECIFICATION MODE

[to be defined consistently with LINE WIDTH SPECIFICATION MODE]

5.5 Control Elements

5.5.1 VDC INTEGER PRECISION

Note -- there is a defect with this element in the CGM standard, in that it is a picture body element whereas VDC Extent, with which it has a very strong interaction, is a Picture Descriptor element. This makes useful effect with this element within the picture body nearly impossible.

Rule:

Permissibility -- per the Model Profile.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

Element is permitted. Shall not be used in the picture body to define a precision which is inconsistent with the VDC Extent.

[Binary: Values are restricted to 16 and 32].
[Character: ...]
[Clear Text: ...]

5.5.2 VDC REAL PRECISION

Rule:

Profile may restrict values.

Permissibility -- per the Model Profile.

Profiles may restrict parameter value if there is cause to do so (note: encoding dependent parameter).

Model:

Permitted.

[Binary: Value is restricted to (1,16,16) and (0,9,23).]
[Character: ...]
[Clear Text: ...]

5.5.3 AUXILIARY COLOUR

Rule:

Profiles shall address whether element is prohibited/permitted.

Model:

Permitted.

5.5.4 TRANSPARENCY

Rule:

Profiles shall address whether element is prohibited/permitted. This shall be consistent with Auxiliary Colour.

Model:

Unrestricted.

5.5.5 CLIP RECTANGLE

Rule:

Profiles shall not prohibit unless allowing the element is inconsistent with other goals of the Profile. Profiles shall address boundary cases such as zero area and area bigger than VDC Extent.

Note -- since ISO 8632 says that objects "inside and on the boundary are drawn", then zero-area does not have the sometimes claimed effect of hiding subsequent primitives -- there will be a visible effect, a dot or line if object intersects the boundary of the degenerate area.

Model:

Permitted. Zero area CLIP RECTANGLE prohibited. Larger than VDC Extent implies clip to VDC Extent.

5.5.6 CLIP INDICATOR

Rule:

Profiles shall address whether element is prohibited/permitted. This shall be consistent with Clip Rectangle.

Model:

Permitted.

═══════ Am.1 ═══════

5.5.7 LINE CLIPPING MODE

Rule:

Profiles shall address whether element is prohibited/permitted.

Profiles may restrict values.

Model:

Permitted; no restrictions.

5.5.8 MARKER CLIPPING MODE

[to be defined consistently with LINE CLIPPING MODE]

5.5.9 EDGE CLIPPING MODE

[to be defined consistently with LINE CLIPPING MODE]

5.5.10 NEW REGION

Rule:

Profiles shall address whether element is prohibited/permitted.

This shall be consistent with BEGIN/END FIGURE. Profiles may restrict the number of occurrences (subregions).

Model:

Permitted; no restrictions.

5.5.11 SAVE PRIMITIVE CONTEXT

Note -- there is debate as to whether this element and the following one should remain in ISO 8632. The effect can be achieved in other ways.

Rule:

Profiles shall only allow this element if there is strong reason to do so within the goals of the profile.

Model:

Prohibited.

5.5.12 RESTORE PRIMITIVE CONTEXT

Rule:

Profile may restrict usage only.

Profiles shall only allow this element if there is strong reason to do so within the goals of the profile.

Model:

Prohibited.

====== Am.3 ======

5.5.13 PROTECTION REGION INDICATOR

Rule:

Profiles shall address whether element is prohibited/permitted.

This shall be consistent with permissibility of BEGIN/END PROTECTION REGION.

Model:

Permitted; no restrictions.

5.5.14 GENERALIZED TEXT PATH MODE

Rule:

Profiles shall address whether element is prohibited/permitted.

Model:

Permitted.

5.5.15 MITRE LIMIT

Rule:

Profiles shall address whether element is prohibited/permitted.

If permitted, Profiles may restrict values.

Model:

Permitted; no restriction.

5.6 Graphical Primitive Elements

5.6.1 POLYLINE

Rule:

The maximum number of points shall be resticted to a specified value or the Profile shall state that the number is unrestricted.

Minimum number of points: per Model Profile.

Geometric degeneracies: per Model Profile.

Model:

Minimum number of points: 2.

Maximum number of points: <= 1024.

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.2 DISJOINT POLYLINE

Rule:

The maximum number of points shall be resticted to a specified value or the Profile shall state that the number is unrestricted.

Minimum number of points: per Model Profile.

Legal values of point count, within extrema: per Model Profile.

Geometric degeneracies: per Model Profile.

Model:

Number of points is even and >= 2.

Number of points is <= 1024.

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.3 POLYMARKER

Rule:

The maximum number of points shall be resticted to a specified value or the profiles shall state that the number is unrestricted.

Minimum number of points: per Model Profile.

Model:

Minimum number of points: 1.

Maximum number of points: 1024.

5.6.4 TEXT

Rule:

Length & content: per "Common rules for graphical text strings".

Model:

Length & content: per "Common rules for graphical text strings".

5.6.5 RESTRICTED TEXT

Rule:

Per common rules for graphical text strings. Semantics & usage: per Model Profile.

Model:

In version 1 & 2 metafiles the meaning of the restricted text shall be as version 3 "boxed" (value 2 of Restricted

Text Method). In version 3, Restricted Text Method must be used if this element is used.

5.6.6 APPEND TEXT

Rule:

Length & content: per "Common rules for graphical text strings".

Model:

Length & content: per "Common rules for graphical text strings".

5.6.7 POLYGON

Rule:

Profile shall restrict the maximum number of points.
Minimum number of points: per Model Profile.

Zero-area degeneracy: per Model Profile.

Model:

The number of points shall be greater than or equal to 3.

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; if points are coincident, the meaning is a dot; if all points are colinear, the meaning is a line.

5.6.8 POLYGON SET

Rule:

Profile shall restrict the maximum number of points.
Minimum number of points: per Model Profile.

Zero-area degeneracy: per Model Profile.

Model:

The number of points shall be greater than or equal to 3.
Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; if points are coincident, the meaning is a dot; if all points are colinear, the meaning is a line.

Sub-polygons, defined as the number of points between start of point list and "close" or between two "closes", shall have al least 3 points.

5.6.9 CELL ARRAY

Rule:

The three defining points must describe a parallelogram of non-zero area.

Profiles must specify maximum values for nx and ny. Minimum values of nx,ny: per Model Profile.

Model:

Permitted. Nx, ny: > 0. Nx, ny: <= 1024.

5.6.10 GENERALIZED DRAWING PRIMITIVE

Rule:

Profiles shall prohibit the use of the GDP, unless a specific subset of registered GDPs is specified, and/or a GDP is unambiguously defined by the profile itself.

Model:

Prohibited.

5.6.11 RECTANGLE

Rule:

Zero-area degeneracy: per Model Profile.

Model:

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; two coincident points define a dot, two colinear points define a line.

5.6.12 CIRCLE

Rule:

Zero-area degeneracy: per Model Profile.

Model:

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; zero radius defines a point.

5.6.13 CIRCULAR ARC 3 POINT

Rule:

Zero-length degeneracy: per Model Profile.

Model:

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies. If 2 points are coincident, the meaning is a line. If three points are coincident, the meaning is a dot. If the three points are colinear, the meaning is a straight line connecting the 3 points.

5.6.14 CIRCULAR ARC 3 POINT CLOSE

Rule:

Zero-area degeneracy: per Model Profile.

Model:

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; see Circular Arc 3 Point.

5.6.15 CIRCULAR ARC CENTRE

Rule:

Zero-length degeneracy: per Model Profile.

Model:

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

A CIRCULAR ARC CENTRE in which the start and end rays are coincident has the meaning of a circle. If the radius = 0, the meaning is a dot.

5.6.16 CIRCULAR ARC CENTRE CLOSE

Rule:

Zero-area degeneracy: per Model Profile.

Model:

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; see Circular Arc Centre.

A CIRCULAR ARC CENTRE in which the start and end rays are coincident has the meaning of a circle and if the close type = 'pie', a radius along the start ray shall be drawn using the current fill and edge attributes.

5.6.17 ELLIPSE

Rule:

Zero-area degeneracy: per Model Profile.

Model:

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies. If the CDPs and centre are coincident, the meaning is a dot; if collinear, the meaning is a line.

5.6.18 ELLIPTICAL ARC

Rule:

Zero-length degeneracy: per Model Profile.

Model:

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies. If the start and end rays are coincident, this element defines the arc of a full ellipse.

5.6.19 ELLIPTICAL ARC CLOSE

Rule:

Zero-area degeneracy: per Model Profile.

Model:

Zero-area geometric degeneracies: per Common Specification for zero-area degeneracies; see Elliptical Arc. If the start and end rays are coincident, this element defines the arc of a full ellipse and if the close type is 'pie' a "radius" is drawn using the current edge attributes.

======= Am.1 =======

5.6.20 CIRCULAR ARC CENTRE REVERSED

Rule:

Zero-length degeneracy: per Model Profile.

Model:

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies. Other: see Circular Arc Centre.

5.6.21 CONNECTING EDGE

Rule:

Zero-length degeneracy: per Model Profile.

Model:

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

======= Am.3 =======

5.6.22 HYPERBOLIC ARC

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.23 PARABOLIC ARC

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.24 **NON-UNIFORM B-SPLINE**

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.25 **NON-UNIFORM RATIONAL B-SPLINE**

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.26 **POLYBEZIER**

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

Zero-length geometric degeneracies: per Common Specification for zero-length degeneracies.

5.6.27 SYMBOL

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

5.6.28 BITONAL TILE

Rule:

[...to be completed].

Zero-length degeneracy: per Model Profile.

Model:

[...to be completed].

5.7 Attribute Elements

[...most of the specifications in this section to be completed yet...]

5.7.1 LINE BUNDLE INDEX

Rule:

Profiles may permit or prohibit. Profile may restrict values for this element.

Model:

Permitted. Values shall be 1..5.

5.7.2 LINE TYPE

Rule:

Profiles may permit or prohibit.

Profile shall restrict values for this element to 1..5, specifically stated sets of registered values, and those negative values which are given clear and precise definitions in the profile (consistent with the "General Principles for Profiles" -- Extensions and Private Values.

[...to be completed...]

Model:

Permitted.

Values limited to 1..5.

5.7.3 LINE WIDTH

Rule:

Profiles may permit or prohibit. Profile may restrict values for this element.

Model:

Permitted. Values unrestricted.

5.7.4 LINE COLOUR

Rule:

Profiles may permit or prohibit. Profiles may restrict values.

Profiles shall enforce rule against Colour Dynamics and rule against Partial Definition of Colours.

Model:

Permitted.

values, Indexed -- Full range of MAXIMUM COLOUR INDEX.
values, Direct -- Full range of COLOUR VALUE EXTENT.

Colour Dynamics and Partial Definition of Colours are prohibited.

5.7.5 MARKER BUNDLE INDEX

[to be defined consistently with LINE BUNDLE INDEX]

5.7.6 MARKER TYPE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.7 MARKER SIZE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.8 MARKER COLOUR

[to be defined consistently with LINE COLOUR]

5.7.9 TEXT BUNDLE INDEX

[to be defined consistently with LINE BUNDLE INDEX]

5.7.10 TEXT FONT INDEX

Rule:

[...to be completed...]

Model:

Index values must be defined.

5.7.11 TEXT PRECISION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.12 CHARACTER EXPANSION FACTOR

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.13 CHARACTER SPACING

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.14 TEXT COLOUR

[to be defined consistently with LINE COLOUR]

5.7.15 CHARACTER HEIGHT

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.16 CHARACTER ORIENTATION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.17 TEXT PATH

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.18 TEXT ALIGNMENT

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.19 CHARACTER SET INDEX

Rule:
[...to be completed...]

Model:

Index values must be defined.

5.7.20 ALTERNATE CHARACTER SET INDEX

Rule:

[...to be completed...]

Model:

Index values must be defined.

5.7.21 FILL BUNDLE INDEX

[to be defined consistently with LINE BUNDLE INDEX]

5.7.22 INTERIOR STYLE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.23 FILL COLOUR

[to be defined consistently with LINE COLOUR]

5.7.24 HATCH INDEX

Rule:

[...to be completed...]

Model:

Index values must be defined.

5.7.25 PATTERN INDEX

Rule:

[...to be completed...]

Model:

Index values must be defined.

5.7.26 EDGE BUNDLE INDEX

[to be defined consistently with LINE BUNDLE INDEX]

5.7.27 EDGE TYPE

[to be defined consistently with LINE TYPE]

5.7.28 EDGE WIDTH

[to be defined consistently with LINE WIDTH]

5.7.29 EDGE COLOUR

[to be defined consistently with LINE COLOUR]

5.7.30 EDGE VISIBILITY

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.31 FILL REFERENCE POINT

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.32 PATTERN TABLE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.33 PATTERN SIZE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.34 COLOUR TABLE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.35 ASPECT SOURCE FLAGS

Rule:

[...to be completed...]

Model:

[...to be completed...]

======= Am.1 =======

5.7.36 PICK IDENTIFIER

Rule:

[...to be completed...]

Model:

[...to be completed...]

======= Am.3 ==

5.7.37 LINE AND EDGE TYPE DEFINITION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.38 LINE CAP

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.39 LINE JOIN

Rule:

[...to be completed...]

v3 profiles prohibit value 1.
v1, v2 profiles define effect equivalent to one of 2..n.

Model:

[...to be completed...]

value 1 is prohibited in v3 metafiles.
In v1, v2 metafiles lines are defined to have butt caps (value 2 of v3).

5.7.40 LINE TYPE CONTINUATION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.41 LINE TYPE INITIAL OFFSET

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.42 TEXT SCORE TYPE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.43 RESTRICTED TEXT TYPE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.44 HATCH STYLE DEFINITION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.45 GEOMETRIC PATTERN

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.46 INTERPOLATED INTERIOR DEFINITION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.47 EDGE CAP

[to be defined consistently with LINE CAP]

5.7.48 EDGE JOIN

[to be defined consistently with LINE JOIN]

5.7.49 EDGE TYPE CONTINUATION

[to be defined consistently with LINE TYPE CONTINUATION]

5.7.50 EDGE TYPE INITIAL OFFSET

[to be defined consistently with LINE TYPE INITIAL OFFSET]

5.7.51 SYMBOL LIBRARY INDEX

Rule:

[...to be completed...]

Model:

Index values must be defined.

5.7.52 SYMBOL COLOUR

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.7.53 SYMBOL SIZE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.8 Escape Elements

5.8.1 ESCAPE

Rule:

[...to be completed...]

Model:
Not permitted

5.9 External Elements

5.9.1 MESSAGE

Rule:

May be restricted in length, action to be taken.
Shall not affect image.

Model:

Limit to max of 32K.

Action required flag is set to "no action"

5.9.2 APPLICATION DATA

Rule:

May not be restricted

Model:

[...to be completed...]

5.10 Segment Elements (Am.1)

5.10.1 Segment Control Elements
5.10.1.1 COPY SEGMENT

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.10.1.2 INHERITANCE FILTER

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.10.1.3 CLIP INHERITANCE

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.10.2 Segment Attribute Elements

5.10.2.1 SEGMENT TRANSFORMATION

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.10.2.2 SEGMENT HIGHLIGHTING

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.10.2.3 SEGMENT DISPLAY PRIORITY

Rule:

[...to be completed...]

Model:

[...to be completed...]

5.10.2.4 SEGMENT PICK PRIORITY

Rule:

[...to be completed...]

Model:

[...to be completed...]

======= **Am.3** =======

7.4 Encoding-dependent Specifications

To be completed.

8. Model Profile

The Model Profile in general does not subset the elements or parameter values of ISO 8632, but contains specifications which must be a feature of every Profile conforming to this annex. The scope of the restrictions of the Model Profile is: to enforce common resolution of ambiguous semantics of ISO 8632; ensure that identical use of identical elements and parameter values has identical meaning under all profiles conforming to this annex; prohibit undefined or ill defined elements or parameter values under profiles conforming to this annex.

The Model Profile:

a) ensures unambiguous semantics for all elements;

b) prohibits all private undefined information;

c) defines specific ranges for the sizes of all elements and values of all parameters.

d) defines specific maxima for the number of simultaneously defined elements of some types (e.g., segments, protection regions) which inherently must be saved by interpreters.

The Model Profile contains: specifications with mandatory values, which shall be incorporated into every conforming Profile; specifications with a basic value supplied in the Model Profile, which value may be changed by valid Profiles with cause to do so; specifications where no value is supplied by the Model Profile but whose value must be defined by conforming Profiles.

The Model Profile is a conformance target for implementations claiming "full CGM implementation". A subset of the Model Profile is a conformance target for any subset implementation claiming "correct CGM implementation". The only notion of "conforming CGM generator" in this standard is: one which puts out only metafiles consistent with the Model Profile, and performs correct semantic and syntactic translation of application graphical objects into CGM elements. The only notion of "conforming CGM interpreter" in this standard is: one which correctly reads any metafile consistent with the Model Profile, and correctly renders or translates its content according to the semantic definitions of this standard.

General Specifications:

Default Bundle Tables (v1):

Colour Dynamics:

[...Define circumstances...are contrary to ISO 8632 philosophy and shall be prohibited by all Profiles.]

Pattern Dynamics:

[...As for colour dynamics].

Full colour definition or no colour definition:

[All Profiles shall require that "all colours which are used shall be defined or none shall." ...this actually belongs in "General Principles" instead of Model Profile.]

Color level: mono, gray, colour.

[Profiles may define these subsets. Model Profile will].

[This section to be completed, consistent with clause 6 and moving relevant material from clause 7].

Issues

- Should the concepts of basic/non-basic, which have been used in much profile work so far, be incorporated into this annex?

- Should all Profiles be forced to have at least a minimal graphical primitive and attribute set?

- Should Profiles address defaults for Version 3 things that don't exist in Version 1 and 2 (e.g., line cap).

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4775

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
FEBRUARY 1992

4. TITLE AND SUBTITLE

Incorporating CALS Requirements into the CGM Standard and the CALS Application Profile--MIL-D-28003

5. AUTHOR(S)

Daniel R. Benigni

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
1/91 through 12/91

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

Defense CALS Executive Office
Pentagon, Room 3D833
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.)

In support of the CALS initiative, one of NIST's major tasks has been to augment the CGM standard with functionality necessary to fulfill CALS requirements for graphical data exchange. This report details the results of NIST representatives' attendance at and monitoring of CGM-related national and international standards meetings. NIST has had significant impact on the development of CGM Amendments 3 and 4 in particular. Amendment 3, which is nearing completion, provides support for advanced two-dimensional drawing capabilities for technical illustration, graphics art quality picture definition, and graphics in technical publishing. Much of the revision to the CGM application profile for CALS, known as MIL-D-28003A, is due to Amendment 3 functionality. Amendment 4, a new standards project, will define rules for profiles and conformance requirements for CGM software components.

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 3; CGM Amendment 4; MIL-D-28003; MIL-D-28003A

13. AVAILABILITY

X	UNLIMITED
	FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
	ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
X	ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
323

15. PRICE
A14

ELECTRONIC FORM