

**Implementation of
EN 28632-2:1994**

**Information
technology —
Computer graphics —
Metafile for the storage
and transfer of picture
description
information —**

**Part 2: Character encoding
(ISO/IEC 8632-2:1992)**

The European Standard EN 28632-2:1994 has the status of a
British Standard

ICS 35.140

NO COPYING WITHOUT BSI PERMISSION EXCEPT AS PERMITTED BY COPYRIGHT LAW

BSI
Documents

AVA

Document Sharing Hub

Committees responsible for this British Standard

The preparation of this British Standard was entrusted by the Information Systems Technology Standards Policy Committee (IST/-) to Technical Committee IST/31, upon which the following bodies were represented:

Department of Trade and Industry [IT Standards Unit (ITD6A)]
Information Systems Committee of the Universities Funding Council
Rutherford Appleton Laboratory

This British Standard, having been prepared under the direction of the Information Systems Technology Standards Policy Committee, was published under the authority of the Standards Board and comes into effect on 15 May 1994

© BSI 01-2000

The following BSI references relate to the work on this standard:
Committee reference IST/31
Draft for comment 91/63009 DC

ISBN 0 580 23341 3

Amendments issued since publication

Amd. No.	Date	Comments
9478	April 1997	Indicated by a sideline in the margin

Contents

	Page
Committees responsible	Inside front cover
National foreword	ii
Foreword	2
Foreword	vii
Text of ISO/IEC 8632-2	1

National foreword

This British Standard reproduces verbatim EN 28632-2:1994 and implements it as the UK national standard. It is a revision of EN 28632-2:1992 and supersedes BS 6945-2:1988 (renumbered as BS EN 28632-2:1992), which is withdrawn.

This British Standard is published under the direction of the Information Systems Technology Standards Policy Committee whose Technical Committee IST/31 has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international committee any enquiries on interpretation, or proposals for change, and keep UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

NOTE International and European Standards, as well as overseas standards, are available from Customer Services, Publications, BSI, Linford Wood, Milton Keynes, MK14 6LE.

Textual errors

When adopting the text of amendment 1 to the International Standard, the textual errors listed below were discovered. They have been reported to ISO.

Pages 65 and 66 The inserted text is clause **11**, not clause **12** as printed.

Page 66 Paragraph beginning “Rules presented as statements” should read “ending with a colon (”:”)” and not “ending with a semicolon (”;”)”.

A British Standard does not purport to include all the necessary provisions of a contract. Users of British Standards are responsible for their correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

Summary of pages

This document comprises a front cover, an inside front cover, pages i and ii, the EN title page, page 2, the ISO/IEC title page, pages ii to viii, pages 1 to 71 and a back cover.

This standard has been updated (see copyright date) and may have had amendments incorporated. This will be indicated in the amendment table on the inside front cover.

Descriptors: Data processing, graphic data processing, information interchange, data handling, data transfer, data representation, data codes, files

English version

Information technology — Computer graphics — Metafile
for the storage and transfer of picture description
information —
Part 2: Character encoding

(includes amendment A1: Rules for profiles)
(ISO/IEC 8632-2:1992 + A1:1994)

Technologies de l'information — Infographie —
Métafichier de stockage et de transfert des
informations de description d'images —
Partie 2: Codage des caractères
(inclut l'amendement A1: Règles pour profils)
(ISO/IEC 8632-2:1992 + A1:1994)

Informationstechnik — Graphische
Datenverarbeitung — Datei für die
Speicherung und Übertragung von
Bildinformationen —
Teil 2: Zeichenkodierung
(enthält Änderung A1: Regeln für Profile)
(ISO/IEC 8632-2:1992 + A1:1994)

This European Standard was approved by CEN on 1994-01-14, amendment A1 was approved by CEN on 1995-11-30. CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the Central Secretariat has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CEN

European Committee for Standardization
Comité Européen de Normalisation
Europäisches Komitee für Normung

Central Secretariat: rue de Stassart 36, B-1050 Brussels

Foreword

On the proposal of the CEN Central Secretariat, the Technical Board decided to submit the International Standard:

Information technology — Computer graphics — Metafile for the storage and transfer of picture description information — Part 2: Character encoding (ISO/IEC 8632-2:1992) to the formal vote.

The result of the formal vote was positive.

For the time being, this document exists only in English and French.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by July 1994, and conflicting national standards shall be withdrawn at the latest by July 1994.

In accordance with the CEN/CENELEC Internal Regulations, the following countries are bound to implement this European Standard: Austria, Belgium, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

Foreword to amendment A1

The text of the Amendment

ISO 8632-2:1992/A1:1994 to the International Standard from ISO/IEC/JTC 1, Information Technology, of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) has been taken over as Amendment to the European Standard by CEN Technical Board.

This Amendment to the European Standard EN 28632-2:1994 shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by June 1996, and conflicting national standards shall be withdrawn at the latest by June 1996.

INTERNATIONAL STANDARD

ISO/IEC
8632-2

Second edition
1992-10-01

Information technology — Computer graphics — Metafile for the storage and transfer of picture description information —

Part 2: Character encoding

*Technologies de l'information — Infographie — Métafichier de stockage
et de transfert des informations de description d'images —*

Partie 2: Codage des caractères



Reference number
ISO/IEC 8632-2:1992(E)

Contents

	Page
Foreword	vii
Introduction	1
1 Scope	1
2 Normative references	1
3 Notational conventions	2
3.1 7-Bit and 8-Bit code tables	2
3.2 Code extension techniques vocabulary	3
3.2.1 C0 sets	3
3.2.2 C1 sets	3
3.2.3 G-sets	3
4 Entering and leaving the metafile environment	5
4.1 Implicitly entering the metafile environment	5
4.2 Designating and invoking the CGM coding environment from ISO 2022	5
5 Method of encoding opcodes	5
5.1 Encoding technique of the basic opcode set	6
5.2 Extension mechanism	6
5.3 Opcode assignments	7
6 Method of encoding parameters	11
6.1 Basic format	11
6.2 Bitstream format	12
6.3 Coding integers	12
6.4 Coding real numbers	13
6.5 Coding VDCs and points	14
6.6 Coding point list parameters	15
6.6.1 Displacement mode	15
6.6.2 Incremental mode	15
6.6.3 Incremental mode encoding	18
6.7 Colour specifiers	19
6.8 Colour lists	19
6.8.1 Normal format (coding type = 0)	20
6.8.2 Bitstream format (coding type = 1)	20
6.8.3 Runlength format (coding type = 2)	20
6.8.4 Runlength bitstream format (coding type = 3)	20
6.8.5 Examples	21
6.9 String parameters	22
6.9.1 Overall string parameter format	22
6.9.2 Bit combinations permitted within string parameters of text elements	22
6.9.3 C0 control within string parameters	23
6.9.4 Using G-sets in string parameters	23
6.10 Enumerated parameters	23
6.11 Index parameters	23
6.12 Data record parameters	23
6.13 Coding VCs and viewport point parameters	24
6.14 Name parameters	24
6.15 Compressed bitstream operands	24
6.16 Structured data record operands	24
6.17 Glyph mapping	24
7 Character substitution	25

	Page
8 Representation of each element	26
8.1 Delimiter elements	27
8.1.1 BEGIN METAFILE	27
8.1.2 END METAFILE	28
8.1.3 BEGIN PICTURE	28
8.1.4 BEGIN PICTURE BODY	28
8.1.5 END PICTURE	28
8.1.6 BEGIN SEGMENT	28
8.1.7 END SEGMENT	28
8.1.8 BEGIN FIGURE	28
8.1.9 END FIGURE	28
8.1.10 BEGIN PROTECTION REGION	28
8.1.11 END PROTECTION REGION	28
8.1.12 BEGIN COMPOUND LINE	28
8.1.13 END COMPOUND LINE	28
8.1.14 BEGIN COMPOUND TEXT PATH	28
8.1.15 END COMPOUND TEXT PATH	28
8.1.16 BEGIN TILE ARRAY	28
8.1.17 END TILE ARRAY	29
8.2 Metafile descriptor elements	29
8.2.1 METAFILE VERSION	29
8.2.2 METAFILE DESCRIPTION	29
8.2.3 VDC TYPE	30
8.2.4 INTEGER PRECISION	30
8.2.5 REAL PRECISION	30
8.2.6 INDEX PRECISION	31
8.2.7 COLOUR PRECISION	31
8.2.8 COLOUR INDEX PRECISION	31
8.2.9 MAXIMUM COLOUR INDEX	31
8.2.10 COLOUR VALUE EXTENT	31
8.2.11 METAFILE ELEMENT LIST	31
8.2.12 METAFILE DEFAULTS REPLACEMENT	32
8.2.13 FONT LIST	32
8.2.14 CHARACTER SET LIST	32
8.2.15 CHARACTER CODING ANNOUNCER	33
8.2.16 NAME PRECISION	33
8.2.17 MAXIMUM VDC EXTENT	33
8.2.18 SEGMENT PRIORITY EXTENT	33
8.2.19 COLOUR MODEL	33
8.2.20 COLOUR CALIBRATION	33
8.2.21 FONT PROPERTIES	34
8.2.22 GLYPH MAPPING	36
8.2.23 SYMBOL LIBRARY LIST	37
8.3 Picture descriptor elements	37
8.3.1 SCALING MODE	37
8.3.2 COLOUR SELECTION MODE	37
8.3.3 LINE WIDTH SPECIFICATION MODE	37
8.3.4 MARKER SIZE SPECIFICATION MODE	38

	Page	
8.3.5	EDGE WIDTH SPECIFICATION MODE	38
8.3.6	VDC EXTENT	38
8.3.7	BACKGROUND COLOUR	38
8.3.8	DEVICE VIEWPORT	38
8.3.9	DEVICE VIEWPORT SPECIFICATION MODE	38
8.3.10	DEVICE VIEWPORT MAPPING	38
8.3.11	LINE REPRESENTATION	39
8.3.12	MARKER REPRESENTATION	39
8.3.13	TEXT REPRESENTATION	39
8.3.14	FILL REPRESENTATION	40
8.3.15	EDGE REPRESENTATION	40
8.3.16	INTERIOR STYLE SPECIFICATION MODE	40
8.3.17	LINE AND EDGE TYPE DEFINITION	41
8.3.18	HATCH STYLE DEFINITION	41
8.3.19	GEOMETRIC PATTERN DEFINITION	41
8.4	Control elements	41
8.4.1	VDC INTEGER PRECISION	41
8.4.2	VDC REAL PRECISION	42
8.4.3	AUXILIARY COLOUR	42
8.4.4	TRANSPARENCY	42
8.4.5	CLIP RECTANGLE	42
8.4.6	CLIP INDICATOR	43
8.4.7	LINE CLIPPING MODE	43
8.4.8	MARKER CLIPPING MODE	43
8.4.9	EDGE CLIPPING MODE	43
8.4.10	NEW REGION	43
8.4.11	SAVE PRIMITIVE CONTEXT	43
8.4.12	RESTORE PRIMITIVE CONTEXT	43
8.4.13	PROTECTION REGION INDICATOR	43
8.4.14	GENERALIZED TEXT PATH MODE	44
8.4.15	MITRE LIMIT	44
8.4.16	TRANSPARENT CELL COLOUR	44
8.5	Graphical primitive elements	44
8.5.1	POLYLINE	44
8.5.2	DISJOINT POLYLINE	44
8.5.3	POLYMARKER	44
8.5.4	TEXT	44
8.5.5	RESTRICTED TEXT	45
8.5.6	APPEND TEXT	45
8.5.7	POLYGON	45
8.5.8	POLYGON SET	45
8.5.9	CELL ARRAY	45
8.5.10	GENERALIZED DRAWING PRIMITIVE	47
8.5.11	RECTANGLE	47
8.5.12	CIRCLE	47
8.5.13	CIRCULAR ARC 3 POINT	47
8.5.14	CIRCULAR ARC 3 POINT CLOSE	47
8.5.15	CIRCULAR ARC CENTRE	47

	Page
8.5.16 CIRCULAR ARC CENTRE CLOSE	48
8.5.17 ELLIPSE	48
8.5.18 ELLIPTICAL ARC	48
8.5.19 ELLIPTICAL ARC CLOSE	48
8.5.20 CIRCULAR ARC CENTRE REVERSED	48
8.5.21 CONNECTING EDGE	49
8.5.22 HYPERBOLIC ARC	49
8.5.23 PARABOLIC ARC	49
8.5.24 NON-UNIFORM B-SPLINE	49
8.5.25 NON-UNIFORM RATIONAL B-SPLINE	49
8.5.26 POLYBEZIER	50
8.5.27 POLYSYMBOL	50
8.5.28 BITONAL TILE	50
8.5.29 TILE	51
8.6 Attribute elements	51
8.6.1 LINE BUNDLE INDEX	51
8.6.2 LINE TYPE	51
8.6.3 LINE WIDTH	52
8.6.4 LINE COLOUR	52
8.6.5 MARKER BUNDLE INDEX	52
8.6.6 MARKER TYPE	52
8.6.7 MARKER SIZE	52
8.6.8 MARKER COLOUR	52
8.6.9 TEXT BUNDLE INDEX	52
8.6.10 TEXT FONT INDEX	52
8.6.11 TEXT PRECISION	52
8.6.12 CHARACTER EXPANSION FACTOR	53
8.6.13 CHARACTER SPACING	53
8.6.14 TEXT COLOUR	53
8.6.15 CHARACTER HEIGHT	53
8.6.16 CHARACTER ORIENTATION	53
8.6.17 TEXT PATH	53
8.6.18 TEXT ALIGNMENT	53
8.6.19 CHARACTER SET INDEX	54
8.6.20 ALTERNATE CHARACTER SET INDEX	54
8.6.21 FILL BUNDLE INDEX	54
8.6.22 INTERIOR STYLE	54
8.6.23 FILL COLOUR	54
8.6.24 HATCH INDEX	54
8.6.25 PATTERN INDEX	54
8.6.26 EDGE BUNDLE INDEX	55
8.6.27 EDGE TYPE	55
8.6.28 EDGE WIDTH	55
8.6.29 EDGE COLOUR	55
8.6.30 EDGE VISIBILITY	55
8.6.31 FILL REFERENCE POINT	55
8.6.32 PATTERN TABLE	55
8.6.33 PATTERN SIZE	55

	Page
8.6.34 COLOUR TABLE	56
8.6.35 ASPECT SOURCE FLAGS	56
8.6.36 PICK IDENTIFIER	57
8.6.37 LINE CAP	57
8.6.38 LINE JOIN	57
8.6.39 LINE TYPE CONTINUATION	57
8.6.40 LINE TYPE INITIAL OFFSET	57
8.6.41 TEXT SCORE TYPE	58
8.6.42 RESTRICTED TEXT TYPE	58
8.6.43 INTERPOLATED INTERIOR	58
8.6.44 EDGE CAP	58
8.6.45 EDGE JOIN	59
8.6.46 EDGE TYPE CONTINUATION	59
8.6.47 EDGE TYPE INITIAL OFFSET	59
8.6.48 SYMBOL LIBRARY INDEX	59
8.6.49 SYMBOL COLOUR	59
8.6.50 SYMBOL SIZE	59
8.6.51 SYMBOL ORIENTATION	60
8.7 Escape elements	60
8.7.1 ESCAPE	60
8.7.2 DOMAIN RING	60
8.8 External elements	60
8.8.1 MESSAGE	60
8.8.2 APPLICATION DATA	60
8.9 Segment elements	61
8.9.1 COPY SEGMENT	61
8.9.2 INHERITANCE FILTER	61
8.9.3 CLIP INHERITANCE	63
8.9.4 SEGMENT TRANSFORMATION	63
8.9.5 SEGMENT HIGHLIGHTING	63
8.9.6 SEGMENT DISPLAY PRIORITY	64
8.9.7 SEGMENT PICK PRIORITY	64
9 Defaults	64
10 Conformance	65
11 Encoding rules, proforma, and Model Profil	65
11.1 General principles	65
11.1.1 Encodings	65
11.1.2 Metafile default	65
11.2 Profile Proforma tables	65
Annex A (normative) Formal grammar	70
Figure 1 —The 7-bit code table	4
Figure 2 —The 8-bit code table	5
Table 1 — Opcodes for metafile elements	7
Table 2 — Huffman code table for incremental mode	17
Table 3 — C0 control set	23
Table 4 — Character substitution	26
Table 5 — Metafile descriptor elements	67
Table 6 — Control elements	69
Table 7 — Escape elements	69

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 8632-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This second edition cancels and replaces the first edition (ISO 8632-2:1987), which has been technically revised.

ISO/IEC 8632 consists of the following parts, under the general title *Information technology — Computer graphics — Metafile for the storage and transfer of picture description information*:

- *Part 1: Functional specification*;
- *Part 2: Character encoding*;
- *Part 3: Binary encoding*;
- *Part 4: Clear text encoding*.

Annex A forms an integral part of this part of ISO/IEC 8632.

Amendment 1 to International Standard ISO/IEC 8632-2:1992 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Sub-committee 24, *Computer graphics and image processing*.

Introduction

0.1 Purpose of the character encoding

The Character Encoding of the Computer Graphics Metafile (CGM) provides a representation of the Metafile syntax intended for situations in which it is important to minimize the size of the metafile or transmit the metafile through character-oriented communications services. The encoding uses compact representation of data that is optimized for storage or transfer between computer systems.

If minimizing the processing overhead is more important than data compaction, an encoding such as the Binary Encoding contained in ISO/IEC 8632-3 may be more appropriate. If human readability is the most important criterion, an encoding such as the Clear Text Encoding in ISO/IEC 8632-4 may be more appropriate.

0.2 Objectives

This encoding was designed with the following objectives:

- a) regular syntax: All elements of the metafile should be encoded in a uniform way so that parsing the metafile is simple;
- b) compactness: The encoding should provide a highly compact metafile, suitable for systems with restricted storage capacity or transfer bandwidth;
- c) extensibility: the encoding should allow for future extensions;
- d) transportability: the encoding should be suitable for use with transport mechanisms designed for character-oriented data based on a standard national character set derived from ISO/IEC 646.

0.3 Metafile characteristics

Each CGM command follows a simple regular syntax. Thus, new commands can be added in a future revision of ISO/IEC 8632 such that existing CGM interpreters can recognize (and ignore) the new commands. Also, new operands can be added to existing commands in the future revision of the standard such that existing CGM interpreters can recognize (and ignore) the additional operands.

Each CGM operand follows a simple regular syntax. Operands are variable in length. This permits small values to be represented by the smallest number of bytes.

A certain range of operand values of standard commands have been reserved for private use; the remaining range is either standardized or reserved for future standardization.

0.4 Relationship to other International Standards

The Character Encoding has been developed in collaboration with the ISO subcommittee responsible for character sets and coding, ECMA, and CEPT. The encoding conforms to the rules for code extension specified in ISO 2022 in the category of complete coding system.

The representation of character data in this part of ISO/IEC 8632 follows the rules of ISO/IEC 646 and ISO 2022.

For certain elements, the CGM defines value ranges as being reserved for registration. The values and their meanings will be defined using the established procedures (see ISO/IEC 8632-1, sub-clause 4.12.).

1 Scope

This part of ISO/IEC 8632 specifies a character encoding of the Computer Graphics Metafile. For each of the elements specified in ISO/IEC 8632-1 an encoding is specified.

This encoding of the Computer Graphics Metafile provides a highly compact representation of the metafile, suitable for applications that require the metafile to be of minimum size and suitable for transmission with character-oriented transmission services.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 8632. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 8632 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO/IEC 646:1991, *Information technology — ISO 7-bit coded character set for information interchange*.

ISO 2022:1986, *Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques*.

ISO 2375:1985, *Data processing — Procedure for registration of escape sequences*.

ISO 6429:1988, *Information processing — Control functions for 7-bit and 8-bit coded character sets*.

ECMA 96, *Graphics Data Syntax for a multiple Workstation Interface*.

CEPT, *Revision of T/CD 6.1 Videotex Presentation Layer Data Syntax*.

3 Notational conventions

3.1 7-Bit and 8-Bit code tables

The bits of the bit combinations of the 7-bit code are identified by b7, b6, b5, b4, b3, b2, and b1, where b7 is the highest-order, or most-significant, bit and b1 is the lowest-order, or least-significant, bit.

The bit combinations may be interpreted to represent integers in the range 0 to 127 in binary notation by attributing the following weights to the individual bits:

Bit:	b7	b6	b5	b4	b3	b2	b1
Weight:	64	32	16	8	4	2	1

In this part of ISO/IEC 8632, the bit combinations are identified by notation of the form *x/y*, where *x* is a number in the range 0 to 7 and *y* is a number in the range 0 to 15. The correspondence between the notations of the form *x/y* and the bit combinations consisting of the bits b7 to b1 is as follows:

- *x* is the number represented by b7, b6, and b5 where these bits are given the weights 4, 2, and 1 respectively;
- *y* is the number represented by b4, b3, b2, and b1 where these bits are given the weights 8, 4, 2, and 1 respectively.

The notations of the form *x/y* are the same as those used to identify code table positions, where *x* is the column number and *y* is the row number.

A 7-bit code table consists of 128 positions arranged in eight columns and sixteen rows. The columns are numbered 0 to 7 and the rows are numbered 0 to 15. Figure 1 shows a 7-bit code table.

An example illustrates the 7-bit code: “1/11” refers to the bit combination in column 1, row 11 of the code table, binary 0011011.

The bits of the bit combinations of the 8-bit code are identified by b8, b7, b6, b5, b4, b3, b2, and b1, where b8 is the highest-order, or most-significant, bit and b1 is the lowest-order, or least-significant, bit.

The bit combinations may be interpreted to represent integers in the range 0 to 255 in binary notation by attributing the following weights to the individual bits:

Bit:	b8	b7	b6	b5	b4	b3	b2	b1
Weight:	128	64	32	16	8	4	2	1

Using these weights, the bit combinations of the 8-bit code are interpreted to represent numbers in the range 0 to 255.

In this part of ISO/IEC 8632, the bit combinations are identified by notation of the form *xx/yy*, where *xx* and *yy* are numbers in the range 00 to 15. The correspondence between the notations of the form *xx/yy* and the bit combinations consisting of the bits b8 to b1 is as follows:

- *xx* is the number represented by b8, b7, b6, and b5 where these bits are given the weights 8, 4, 2, and 1 respectively;
- *yy* is the number represented by b4, b3, b2, and b1 where these bits are given the weights 8, 4, 2, and 1 respectively.

The notations of the form *xx/yy* are the same as those used to identify code table positions, where *xx* is the column number and *yy* is the row number. An 8-bit code table consists of 256 positions arranged in sixteen columns and sixteen rows. The columns and rows are numbered 00 to 15. Figure 2 shows an 8-bit code table.

An example illustrates the 8-bit code: 04/01 represents the 8-bit byte 01000001, whereas 4/1 represents the 7-bit byte 1000001.

3.2 Code extension techniques vocabulary

In describing the characters that may occur within string parameters, certain terms imported from other standards (e.g., ISO 2022) are useful. In the context of the CGM, these terms, and the concepts to which they refer, apply only within the string parameters of the TEXT, APPEND TEXT, and RESTRICTED TEXT metafile elements.

3.2.1 C0 sets

A C0 set is a set of 30 control characters represented in a 7-bit code by 0/0 to 1/15, except 0/14 and 0/15 which shall be unused, and in an 8-bit code by 00/00 to 01/15, except 00/14 and 00/15 which shall be unused. C0 sets occupy columns 0 and 1 of a 7-bit code table or columns 00 and 01 of an 8-bit code table. The meanings of C0 controls within string parameters are described in 6.9.3.

3.2.2 C1 sets

A C1 set is a set of up to 32 control characters represented by bit combinations 08/00 to 09/15 in an 8-bit code. C1 sets occupy columns 08 and 09 of the 8-bit code table. In a 7-bit code the C1 control functions are represented by 2-byte escape sequences. This CGM encoding reserves the bit combinations 9/8 and 9/12 (ESC 5/8 and ESC 5/12 in a 7-bit environment, ESC = 1/11); these shall not be part of the content of string parameters. Other C1 control characters from other standards, such as ISO 6429, may be used within string parameters by agreement between the interchanging parties.

3.2.3 G-sets

The G-sets (G0, G1, G2, G3) are coded character sets of 94 or 96 characters. CHARACTER SET INDEX designates which character set is to be the G0 set. ALTERNATE CHARACTER SET INDEX designates a character set to be used as both the G1 and G2 sets. The G-sets may be “invoked into” (caused to occupy) columns 2 through 7 of a 7-bit code table, or columns 02 through 07 and 10 through 15 of an 8-bit code table. This encoding of the CGM uses the G0 and G1/G2 sets within string parameters. The G3 set may be used within the string parameters of conforming metafiles; this requires selection of the extended 7-bit or extended 8-bit mode in the CHARACTER CODING ANNOUNCER. The CGM does not provide an element to explicitly designate the G3 sets; this may be done within a text string in accordance with ISO 2022, or by other means agreed upon by the interchanging parties.

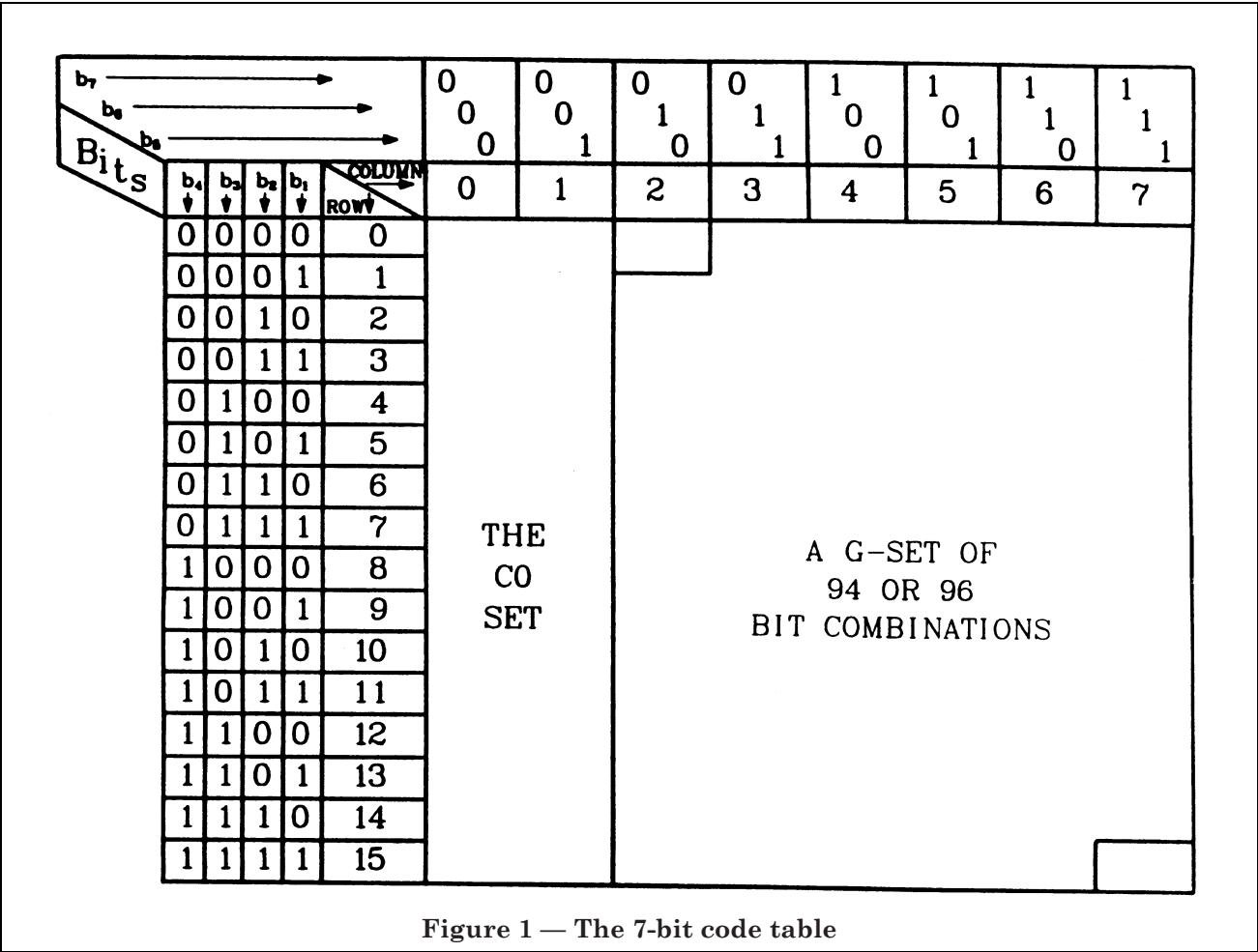


Figure 1 — The 7-bit code table

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
0			02/0								10/0					
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15								07/15							15/15	

Figure 2 — The 8-bit code table

4 Entering and leaving the metafile environment

4.1 Implicitly entering the metafile environment

The CGM coding environment may be entered implicitly, by agreement between the interchanging parties. This is suitable only if there is not to be any interchange with services using other coding techniques.

4.2 Designating and invoking the CGM coding environment from ISO 2022

For interchange with services using the code extension techniques of ISO 2022, the CGM coding environment shall be designated and invoked from ISO 2022 environment by the following escape sequence:

ESC 2/5 F

where ESC is the bit combination 1/11, and F refers to a bit combination that will be assigned by the ISO Registration Authority for ISO 2375.

The first bit combination occurring after this escape sequence will then represent the opcode of a CGM metafile element.

After the end of one or more metafiles (i.e., after the END METAFILE element) or between pictures (i.e., after the END PICTURE element), the following escape sequence may be used to return to the ISO 2022 coding environment:

ESC 2/5 4/0

This not only returns to the ISO 2022 coding environment, but also restores the designation and invocation of coded character sets to the state that existed prior to entering the CGM coding environment with the ESC 2/5 F sequence. (The terms “designation” and “invocation” are defined in ISO 2022.)

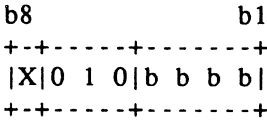
5 Method of encoding opcodes

Each metafile element is composed of one opcode and parameters as required. The opcodes are coded as a sequence of bit combinations from columns 2 and 3 of the code chart. The encoding technique supplies:

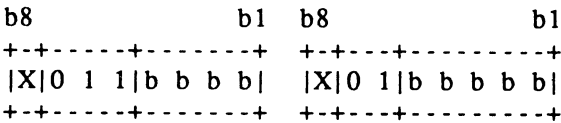
- the basic opcode set;
- extension opcode sets.

5.1 Encoding technique of the basic opcode set

The basic opcode set consists of single-byte and double-byte opcodes. Single-byte opcodes are from column 2 of the code chart. Bits b4 to b1 are used to encode the opcode. The format is as follows:



The “X” bit (bit b8) is the parity bit (or omitted bit) in a 7-bit environment. In an 8-bit environment it is 0. For double-byte opcodes the first byte is from column 3 and the second byte is from column 2 or 3 of the code chart. Bits b4 to b1 of the first byte and bits b5 to b1 of the second byte are used to encode the opcode:



The bit combination 3/15, the EXTEND OPCODE SPACE (EOS) allows extension of the basic opcode space (see 5.2).

The basic opcode set, supplied by this encoding technique consists of 496 opcodes, being:

- 16 single-byte opcodes (from column 2);
- 15 × 32 = 480 double-byte opcodes (first byte from column 3 except bit combination 3/15, second byte from column 2 or 3).

5.2 Extension mechanism

The basic opcode set can be extended with an unlimited number of extension opcode sets by means of the EXTEND OPCODE SPACE code (EOS, 3/15).

The N-th extension opcode set consists of opcodes of the basic opcode set, prefixed with n times the code EOS. The three possible formats of an opcode from the N-th extension opcode set are

Opcode format	Extension codes	Basic opcode set codes
1	<EOS> . . . <EOS> ----- n t i m e s	<2/x>
2	<EOS> . . . <EOS> ----- n t i m e s	<3/y> <2/z>
3	<EOS> . . . <EOS> ----- n t i m e s	<3/y> <3/z>

<EOS> = 3/15

x = 0,1,...,15

y = 0,1,...,14

z = 0,1,...,15

n = 0,1,.....

n = 0 selects the basic opcode set.

n = 1 selects the first extension opcode set.

n = N selects the N-th extension opcode set.

The number of opcodes supplied by this encoding technique (basic opcode set plus extension opcode sets) is $496 \cdot (n + 1)$, where n is the number of extension sets. (Each extension set has 496 opcodes — 16 single-byte opcodes plus 480 double-byte codes.)

5.3 Opcode assignments

Table 1 lists the opcode assignments for the CGM elements. All opcodes are from the basic opcode set. They are organized as follows: single-byte opcodes are assigned to Graphical Primitive elements except for some of the Circular and Elliptical output elements, where double-byte opcodes with combination 3/4 for the first byte are assigned. All other metafile elements have double-byte opcodes with the following bit combinations for the first byte:

3/0	for Delimiter Elements
3/1	for Metafile Descriptor Elements
3/2	for Picture Descriptor Elements
3/3	for Control Elements
3/5 and 3/6	for Attribute Elements
3/7	for Escape and External Elements
3/8	for Segment Control Elements and Segment Attribute Elements

Table 1 — Opcodes for metafile elements

Opcode	7-Bit coding		8-Bit coding	
BEGIN METAFILE opcode	3/0	2/0	03/0	02/0
END METAFILE opcode	3/0	2/1	03/0	02/1
BEGIN PICTURE opcode	3/0	2/2	03/0	02/2
BEGIN PICTURE BODY opcode	3/0	2/3	03/0	02/3
END PICTURE opcode	3/0	2/4	03/0	02/4
BEGIN SEGMENT opcode	3/0	2/5	03/0	02/5
END SEGMENT opcode	3/0	2/6	03/0	02/6
BEGIN FIGURE opcode	3/0	2/7	03/0	02/7
END FIGURE opcode	3/0	2/8	03/0	02/8
BEGIN PROTECTION REGION opcode	3/0	2/12	03/0	02/12
END PROTECTION REGION opcode	3/0	2/13	03/0	02/13
BEGIN COMPOUND LINE opcode	3/0	2/14	03/0	02/14
END COMPOUND LINE opcode	3/0	2/15	03/0	02/15
BEGIN COMPOUND TEXT PATH opcode	3/0	3/0	03/0	03/0
END COMPOUND TEXT PATH opcode	3/0	3/1	03/0	03/1
BEGIN TILE ARRAY opcode	3/0	3/2	03/0	03/2
END TILE ARRAY opcode	3/0	3/3	03/0	03/3
METAFILE VERSION opcode	3/1	2/0	03/1	02/0
METAFILE DESCRIPTION opcode	3/1	2/1	03/1	02/1
VDC TYPE opcode	3/1	2/2	03/1	02/2
INTEGER PRECISION opcode	3/1	2/3	03/1	02/3
REAL PRECISION opcode	3/1	2/4	03/1	02/4
INDEX PRECISION opcode	3/1	2/5	03/1	02/5
COLOUR PRECISION opcode	3/1	2/6	03/1	02/6
COLOUR INDEX PRECISION opcode	3/1	2/7	03/1	02/7
MAXIMUM COLOUR INDEX opcode	3/1	2/8	03/1	02/8
COLOUR VALUE EXTENT opcode	3/1	2/9	03/1	02/9
METAFILE ELEMENT LIST opcode	3/1	2/10	03/1	02/10
BEGIN METAFILE DEFAULTS REPLACEMENT opcode	3/1	2/11	03/1	02/11
END METAFILE DEFAULTS REPLACEMENT opcode	3/1	2/12	03/1	02/12
FONT LIST opcode	3/1	2/13	03/1	02/13
CHARACTER SET LIST opcode	3/1	2/14	03/1	02/14

Table 1 — Opcodes for metafile elements

Opcode	7-Bit coding		8-Bit coding	
CHARACTER CODING ANNOUNCER opcode	3/1	2/15	03/1	02/15
NAME PRECISION opcode	3/1	3/0	03/1	03/0
MAXIMUM VDC EXTENT opcode	3/1	3/1	03/1	03/1
SEGMENT PRIORITY EXTENT opcode	3/1	3/2	03/1	03/2
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
SCALING MODE opcode	3/2	2/0	03/2	02/0
COLOUR SELECTION MODE opcode	3/2	2/1	03/2	02/1
LINE WIDTH SPECIFICATION MODE opcode	3/2	2/2	03/2	02/2
MARKER SIZE SPECIFICATION MODE opcode	3/2	2/3	03/2	02/3
EDGE WIDTH SPECIFICATION MODE opcode	3/2	2/4	03/2	02/4
VDC EXTENT opcode	3/2	2/5	03/2	02/5
BACKGROUND COLOUR opcode	3/2	2/6	03/2	02/6
DEVICE VIEWPORT opcode	3/2	2/7	03/2	02/7
DEVICE VIEWPORT SPECIFICATION MODE opcode	3/2	2/8	03/2	02/8
DEVICE VIEWPORT MAPPING opcode	3/2	2/9	03/2	02/9
LINE REPRESENTATION opcode	3/2	2/10	03/2	02/10
MARKER REPRESENTATION opcode	3/2	2/11	03/2	02/11
TEXT REPRESENTATION opcode	3/2	2/12	03/2	02/12
FILL REPRESENTATION opcode	3/2	2/13	03/2	02/13
EDGE REPRESENTATION opcode	3/2	2/14	03/2	02/14
INTERIOR STYLE SPECIFICATION MODE opcode	3/2	2/15	03/2	02/15
LINE AND EDGE TYPE DEFINITION opcode	3/2	3/0	03/2	03/0
HATCH STYLE DEFINITION opcode	3/2	3/1	03/2	03/1
GEOMETRIC PATTERN DEFINITION opcode	3/2	3/2	03/2	03/2
VDC INTEGER PRECISION opcode	3/3	2/0	03/3	02/0
VDC REAL PRECISION opcode	3/3	2/1	03/3	02/1
AUXILIARY COLOUR opcode	3/3	2/2	03/3	02/2
TRANSPARENCY opcode	3/3	2/3	03/3	02/3
CLIP RECTANGLE opcode	3/3	2/4	03/3	02/4
CLIP INDICATOR opcode	3/3	2/5	03/3	02/5
LINE CLIPPING MODE opcode	3/3	2/6	03/3	02/6
MARKER CLIPPING MODE opcode	3/3	2/7	03/3	02/7
EDGE CLIPPING MODE opcode	3/3	2/8	03/3	02/8
NEW REGION opcode	3/3	2/9	03/3	02/9
SAVE PRIMITIVE CONTEXT opcode	3/3	2/10	03/3	02/10
RESTORE PRIMITIVE CONTEXT opcode	3/3	2/11	03/3	02/11
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
TRANSPARENT CELL COLOUR opcode	3/3	3/3	03/3	03/3
POLYLINE opcode	2/0		02/0	
DISJOINT POLYLINE opcode	2/1		02/1	
POLYMARKER opcode	2/2		02/2	
TEXT opcode	2/3		02/3	

Table 1 — Opcodes for metafile elements

Opcode	7-Bit coding		8-Bit coding	
RESTRICTED TEXT opcode	2/4		02/4	
APPEND TEXT opcode	2/5		02/5	
POLYGON opcode	2/6		02/6	
POLYGON SET opcode	2/7		02/7	
CELL ARRAY opcode	2/8		02/8	
GENERALIZED DRAWING PRIMITIVE opcode	2/9		02/9	
RECTANGLE opcode	2/10		02/10	
CIRCLE opcode	3/4	2/0	03/4	02/0
CIRCULAR ARC 3 POINT opcode	3/4	2/1	03/4	02/1
CIRCULAR ARC 3 POINT CLOSE opcode	3/4	2/2	03/4	02/2
CIRCULAR ARC CENTRE opcode	3/4	2/3	03/4	02/3
CIRCULAR ARC CENTRE CLOSE opcode	3/4	2/4	03/4	02/4
ELLIPSE opcode	3/4	2/5	03/4	02/5
ELLIPTICAL ARC opcode	3/4	2/6	03/4	02/6
ELLIPTICAL ARC CLOSE opcode	3/4	2/7	03/4	02/7
CIRCULAR ARC CENTRE REVERSED opcode	3/4	2/8	03/4	02/8
CONNECTING EDGE opcode	3/4	2/9	03/4	02/9
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
POLYSYMBOL 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 BUNDLE INDEX opcode	3/5	2/0	03/5	02/0
LINE TYPE opcode	3/5	2/1	03/5	02/1
LINE WIDTH	3/5	2/2	03/5	02/2
LINE COLOUR opcode	3/5	2/3	03/5	02/3
MARKER BUNDLE INDEX opcode	3/5	2/4	03/5	02/4
MARKER TYPE opcode	3/5	2/5	03/5	02/5
MARKER SIZE opcode	3/5	2/6	03/5	02/6
MARKER COLOUR opcode	3/5	2/7	03/5	02/7
TEXT BUNDLE INDEX opcode	3/5	3/0	03/5	03/0
TEXT FONT INDEX opcode	3/5	3/1	03/5	03/1
TEXT PRECISION opcode	3/5	3/2	03/5	03/2
CHARACTER EXPANSION FACTOR opcode	3/5	3/3	03/5	03/3
CHARACTER SPACING opcode	3/5	3/4	03/5	03/4
TEXT COLOUR opcode	3/5	3/5	03/5	03/5
CHARACTER HEIGHT opcode	3/5	3/6	03/5	03/6
CHARACTER ORIENTATION opcode	3/5	3/7	03/5	03/7
TEXT PATH opcode	3/5	3/8	03/5	03/8
TEXT ALIGNMENT opcode	3/5	3/9	03/5	03/9
CHARACTER SET INDEX opcode	3/5	3/10	03/5	03/10
ALTERNATE CHARACTER SET INDEX opcode	3/5	3/11	03/5	03/11
FILL BUNDLE INDEX opcode	3/6	2/0	03/6	02/0
INTERIOR STYLE opcode	3/6	2/1	03/6	02/1
FILL COLOUR opcode	3/6	2/2	03/6	02/2
HATCH INDEX opcode	3/6	2/3	03/6	02/3

Table 1 — Opcodes for metafile elements

Opcode	7-Bit coding		8-Bit coding	
PATTERN INDEX opcode	3/6	2/4	03/6	02/4
EDGE BUNDLE INDEX opcode	3/6	2/5	03/6	02/5
EDGE TYPE opcode	3/6	2/6	03/6	02/6
EDGE WIDTH opcode	3/6	2/7	03/6	02/7
EDGE COLOUR opcode	3/6	2/8	03/6	02/8
EDGE VISIBILITY opcode	3/6	2/9	03/6	02/9
FILL REFERENCE POINT opcode	3/6	2/10	03/6	02/10
PATTERN TABLE opcode	3/6	2/11	03/6	02/11
PATTERN SIZE opcode	3/6	2/12	03/6	02/12
COLOUR TABLE opcode	3/6	3/0	03/6	03/0
ASPECT SOURCE FLAGS opcode	3/6	3/1	03/6	03/1
PICK IDENTIFIER opcode	3/6	3/2	03/6	03/2
LINE CAP opcode	3/5	2/8	03/5	02/8
LINE JOIN opcode	3/5	2/9	03/5	02/9
LINE TYPE CONTINUATION opcode	3/5	2/10	03/5	02/10
LINE TYPE INITIAL OFFSET opcode	3/5	2/11	03/5	02/11
TEXT SCORE TYPE opcode	3/5	2/12	03/5	02/12
RESTRICTED TEXT TYPE opcode	3/5	2/13	03/5	02/13
INTERPOLATED INTERIOR opcode	3/5	2/14	03/5	02/14
EDGE CAP opcode	3/5	2/15	03/5	02/15
EDGE JOIN opcode	3/6	2/13	03/6	02/13
EDGE TYPE CONTINUATION opcode	3/6	2/14	03/6	02/14
EDGE TYPE INITIAL OFFSET opcode	3/6	2/15	03/6	02/15
SYMBOL LIBRARY INDEX opcode	3/6	3/3	03/6	03/3
SYMBOL COLOUR opcode	3/6	3/4	03/6	03/4
SYMBOL SIZE opcode	3/6	3/5	03/6	03/5
SYMBOL ORIENTATION opcode	3/6	3/6	03/6	03/6
ESCAPE opcode	3/7	2/0	03/7	02/0
DOMAIN RING opcode	3/7	3/0	03/7	03/0
MESSAGE opcode	3/7	2/1	03/7	02/1
APPLICATION DATA opcode	3/7	2/2	03/7	02/2
COPY SEGMENT opcode	3/8	2/0	03/8	02/0
INHERITANCE FILTER opcode	3/8	2/1	03/8	02/1
CLIP INHERITANCE opcode	3/8	2/2	03/8	02/2
SEGMENT TRANSFORMATION opcode	3/8	2/3	03/8	02/3
SEGMENT HIGHLIGHTING opcode	3/8	2/4	03/8	02/4
SEGMENT DISPLAY PRIORITY opcode	3/8	2/5	03/8	02/5
SEGMENT PICK PRIORITY opcode	3/8	2/6	03/8	02/6
NOTE There is no no-op element in the character encoding. However, the character substitution mechanism allows generators to achieve this effect.				

6 Method of encoding parameters

The parameter part of a CGM element may contain one or more parameters, each parameter consisting of one or more bytes.

All parameters are coded in columns 4 through 7. (However, the coded representation of a “string” parameter may include bit combinations from other columns of the code table — see the description of string parameters in 6.9). The general format of a parameter byte is

```

b8          b1
+--+-----+
|X|1|b b b b|
+--+-----+

```

The “X” (bit b8) is the parity bit (or omitted bit) in a 7-bit environment. In an 8-bit environment it is 0. Bit b7 is the parameter flag.

Except for “string”, “string fixed”, and “data record” parameters all parameters are encoded using one or both of two formats, Basic format or Bitstream format.

6.1 Basic format

Each Basic format parameter is coded as a sequence of one or more bytes, structured as follows:

```

b8          b1
+--+-----+
|X|1|e|s|b b b b|  first byte
+--+-----+
.
.
.
b8          b1
+--+-----+
|X|1|e|b b b b b|  last byte
+--+-----+

```

The “X” (bit b8) is the parity bit (or omitted bit) in a 7-bit environment. In an 8-bit environment it is 0. Bit b7 is the parameter flag.

“e” (b6 of each byte) is the extension flag. For single byte parameters, the extension flag is 0. In multi-byte parameters, the extension flag is 1 in all bytes except the last byte, where it is 0.

Bits b5 through b1 are the data bits of the parameter.

“s” is the sign bit; if equal to 0 then the integer is non-negative and if equal to 1 then the integer is negative. The number zero shall always be coded as “plus zero” 4/0. The “minus zero” coding is reserved for special usage (see 6.6.3).

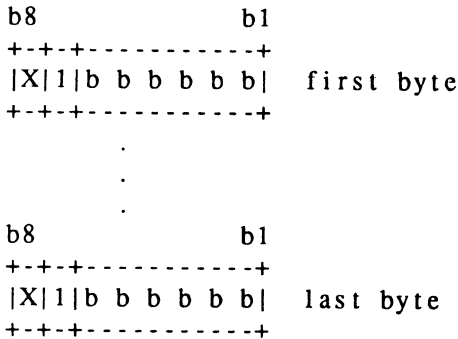
The Basic format is used to encode

- a) enumerated types (E);
- b) colour indices (CI);
- c) colour components (CCO);
- d) indices other than colour indices (IX);
- e) integers (I);
- f) real numbers (R);
- g) non-incremental coordinates.
- h) fixed-precision 8-bit unsigned integers — octets (UI8);
- i) fixed-precision 32-bit unsigned integers (UI32).

The most significant part of the parameter is coded in the first byte. The least significant part of the parameter is coded in the last byte.

6.2 Bitstream format

Each Bitstream format parameter is encoded as a sequence of one or more bytes, structured as follows:



The “X” is the parity bit (or omitted bit) in a 7-bit environment. In an 8-bit environment it is 0. Bit b7 is the parameter flag.

Bits b6 through b1 are the data bits of the parameter.

The Bitstream format is used to encode:

- a) incremental mode coordinates (see 6.6.2);
- b) colour direct (see 6.8);
- c) colour index lists (see 6.8).
- d) the compressed bitstream (BS) data type (see 6.15).

Bitstream data are packed in consecutive databits starting from high-numbered bits to lower-numbered bits of the first byte for the most significant part of the bitstream data.

The end of a Bitstream format parameter cannot be derived from the Bitstream format itself (the format is not self-delimiting). Instead,

- for incremental mode coordinates, the end of the data (which identifies the end of the Bitstream format parameter) is identified by the <End of Block> code;
- for colour index lists, the number of bits needed to encode the colour index list (which identifies the end of the Bitstream format parameter) is set by the COLOUR INDEX PRECISION element, or by the *local colour precision* parameter (for those elements which contain such a parameter);
- for colour direct data, the number of bits needed to encode the data (which identifies the end of the Bitstream format operand) is set by the COLOUR PRECISION element, or by the *local colour precision* parameter (for those elements which contain such a parameter).
- for compressed binary colour specifiers of Tile Array elements, the SOS/ST delimiters that are also used for String operands are used to delimit the Bitstream operand.

6.3 Coding integers

Integers are coded as sequences of bytes in the range from 4/0 to 7/15 in the Basic format. If a byte is from columns 4 or 5 of the code table, it is either the last byte in the integer’s coded representation or it is a single-byte integer. A multi-byte integer begins with a byte from columns 6 or 7 of the code table.

The structure of integer parameters is as illustrated for Basic format (see 6.1). “bbb...” are bits representing the magnitude of the integer. The most significant part of the parameter is coded in the first byte. The least significant part of the parameter is coded in the last byte.

Any integer can be coded with leading most significant bits which are all zero. For example, 4/3 and 6/0 6/0 4/3 are both valid codings for the integer “+ 3”; however, efficient metafile generators should avoid such redundant codings.

The size of integer parameters is limited by the current INTEGER PRECISION value.

Integers in the range of – 15 to + 15 can be coded as single bytes.

- (integer: + 1) = 4/1

(integer: – 1) = 5/1

(integer: + 15) = 4/15

(integer: – 15) = 5/15