

© British Standards Institution. No part of this publication may be photocopied or otherwise reproduced without the prior permission in writing of BSI

Draft for Development

Computer graphics: Metafile for the storage and transfer of picture description information

Part 4. Clear text encoding (ISO/DIS 8632/4)

Projet à développer

Infographie

Métafichier graphique pour la communication et l'archivage d'information graphique

Partie 4. Codage du texte

Vornorm

Graphische Datenverarbeitung

Metadatei für die Speicherung und Übertragung von Informationen zur Bildbeschreibung

Teil 4. Klartextkodierung

Contents

	Page		Page
National foreword	2	5.2 Separators	9
Committees responsible	Back cover	5.3 Encoding of parameter types	10
Draft for Development		5.4 Forming names	13
0 Introduction	3	6 Encoding the CGM elements	17
0.1 Purpose of the clear text encoding	3	6.1 The ELEMENT production	17
0.2 Primary objectives	3	6.2 Encoding delimiter elements	18
0.3 Secondary objectives	3	6.3 Encoding metafile descriptor elements	19
0.4 Relationship to other standards	3	6.4 Encoding picture descriptor elements	21
0.5 Status of annexes	3	6.5 Encoding control elements	22
1 Scope and field of application	3	6.6 Encoding graphical primitive elements	23
1.1 Scope	4	6.7 Encoding attribute elements	28
1.2 Field of application	4	6.8 Encoding escape elements	32
2 References	4	6.9 Encoding external elements	32
3 Notational conventions	5	7 Clear text encoding defaults	34
4 Entering and leaving the metafile environment	6	8 Clear text encoding conformance	35
4.1 Generic clear text and instantiations	7	Annexes	
4.2 Implicitly entering the metafile environment	7	A Clear text encoding-dependent formal grammar	38
4.3 Invoking the CGM coding environment from ISO 2022	7	B Clear text encoding example	39
5 Metafile format	8		
5.1 Character repertoire	8		

This publication is not to be regarded as a British Standard.

It is being issued in the Draft for Development series of publications and is of a provisional nature because the subject matter is in the process of ongoing development within the International Standards Committee (ISO/TC 97/SC 21 Information Retrieval, Transfer and Management for Open Systems Interconnection).

It should be applied on this provisional basis, so that information and experience of its practical application may be obtained.

A review of this Draft for Development will be carried out not later than two years after its publication. Notification

of the start of the review period, with a request for the submission of comments from users of this Draft for Development will be made in an announcement in the appropriate issue of *BSI News*. Observations which it is felt should receive attention before the official call for comments will be welcomed.

According to the replies received, the responsible BSI Committee will judge whether the Draft for Development can be converted into a British Standard or what other action should be taken.

National foreword

This Part of DD 107 has been prepared under the direction of the Office and Information Standards Committee.

The primary objective of this Part of DD 107 is to give advance guidance to product and systems designers concerned with computer graphics.

The text of this Part of DD 107 is identical with Draft International Standard ISO/DIS 8632/4-1985, issued by the International Organization for Standardization (ISO).

For ease of reproduction, the ISO text has been reproduced unchanged.

The UK is actively participating through BSI Technical Committee IST/21* in the development of the ISO International Standard. The Technical Committee will review any agreed amendment or addenda to the ISO text and will, where appropriate, update the text of this Draft for Development by means of an amendment or revision. When the ISO draft reaches an appropriate level of agreement and stability, this Part of DD 107 may then be converted to a British Standard.

The four Parts of ISO/DIS 8632 are being published as corresponding Parts of DD 107, as follows.

ISO/DIS 8632/1-1985 is identical with
DD 107 : Part 1 Functional specification

ISO/DIS 8632/2-1985 is identical with
DD 107 : Part 2 Character encoding

ISO/DIS 8632/3-1985 is identical with
DD 107 : Part 3 Binary encoding

ISO/DIS 8632/4-1985 is identical with
DD 107 : Part 4 Clear text encoding

All four Parts are currently at the Draft International Standard stage.

A brochure giving details of existing and projected British Standards publications for Open Systems Interconnection

(OSI) may be obtained from the Sales Department, BSI Linford Wood, Milton Keynes, MK14 6LE, at no charge.

Press releases for new OSI, data communications and computer graphics British Standards publications will be issued to the computing press, and details of all new publications will be recorded in full in *BSI News*.

Terminology and conventions. The text of the Draft International Standard has been approved as suitable for publication as a Draft for Development without deviation. Some terminology and certain conventions are not identical with those used in Drafts for Development; attention is drawn especially to the following.

Wherever the word 'Standard' appears, referring to this Draft for Development, it should be read as 'Draft for Development'.

Wherever page numbers are quoted, they are ISO page numbers.

Cross-references

International Standard	Corresponding British Standard
ISO 646-1983	BS 4730 : 1985 Specification for the UK 7 bit data code (Identical, exercising national options)
ISO 7942-1985	BS 6390 : 1985 Specification for a set of functions for computer graphics programming, the Graphical Kernel System (GKS) (Identical)
ISO 2022-1982, to which reference is made in the text, is currently being revised, and this revision may be implemented as a British Standard. BS 4953, which was technically equivalent to ISO 2022-1973, was withdrawn in June 1984.	

*Previously OIS/121.

0 Introduction

0.1 Purpose of the Clear Text Encoding

The Clear Text Encoding of the Computer Graphics Metafile (CGM) provides a representation of the Metafile syntax that is easy to type, edit and read. It allows a metafile to be edited with any standard text editor, using the internal character code of the host computer system.

0.2 Primary Objectives

- a. **HUMAN EDITABLE:** The Clear Text Encoding should be able to be hand edited or, if desired, hand constructed.
- b. **HUMAN-FRIENDLY:** The Clear Text Encoding should be easy and natural for people to read and edit. Although what is easiest and most natural is a subjective judgment that varies among users, contributing factors such as ease of recognition, ease of remembering, avoidance of ambiguity, and prevention of mistyping have all been considered.
- c. **MACHINE-READABLE:** The Clear Text Encoding should be able to be parsed by software.
- d. **Suitable for USE IN A WIDE VARIETY OF EDITORS:** The Clear Text Encoding should not have any features that make it difficult to edit in normal text editors.
- e. **Facilitate INTERCHANGE BETWEEN DIVERSE SYSTEMS:** The Clear Text Encoding should be encoded in such a way as to maximize the set of systems which can utilize it. No assumptions should be made as to word size or arithmetic modes used to interpret the metafile.
- f. **Use STANDARDIZED ABBREVIATIONS** as much as possible. Where language encoding of other graphics standards have established standard abbreviations, or where common practice in the data processing and graphics industries has established well known abbreviations, these abbreviations are used. In accordance with the principle of "least astonishment", this approach should minimize the time needed to learn to use this encoding.

0.3 Secondary Objectives

Because other CGM encodings are targeted toward CPU efficiency (CGM Binary Encoding) and information density (CGM Character Encoding), these objectives are considered of secondary importance for the CGM Clear Text Encoding.

0.4 Relationship to Other Standards

The set of characters required to implement the Clear Text Encoding is a subset of those included in national versions of ISO 646. Any character set that can be mapped to and from that subset may be used to implement the encoding.

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 part 1, 4.11.)

0.5 Status of Annexes

The annexes do not form an integral part of the Standard but are included to provide extra information and explanation.

1 Scope and Field of Application

1.1 Scope

This part of this Standard specifies a clear text encoding of the Computer Graphics Metafile. For each of the elements specified in part 1 of this Standard, a clear text encoding is specified. Allowed abbreviations are specified. The overall format of the metafile and the means by which comments may be interspersed in the metafile is specified.

1.2 Field of Application

This encoding of the CGM allows metafiles to be created and maintained in a form which is simple to type, easy to edit and convenient to read.

[ISO page 2]

This publication is a Draft for Development and is therefore not to be regarded as a British Standard. See page 1.

2 References

The publications referred to in this part of the Standard are as follows.

- ISO 646 7-bit coded character set for information processing interchange.
- ISO 2022 Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques.

[ISO page 3]

This publication is a Draft for Development and is therefore not to be regarded as a British Standard. See page 1.

Documents
Document Sharing Hub

3 Notational Conventions

Unbracketed strings are terminals of this grammar which appear exactly, subject to the notes on case and null characters given below.

Bracketed strings are either non-terminals (with further productions given), character symbol names (such as COMMA), or parameters of the CGM element in the form $\langle x:y \rangle$ (see part 1 of the Standard for further explanation of these items).

"::=" is read as "becomes" or "is realized as".

$\langle \dots \rangle^*$	= star closure (0 or more occurrences).
$\langle \dots \rangle^+$	= plus closure (1 or more occurrences).
$\langle \dots \rangle^o$	= optional (exactly 0 or 1 occurrences).
$\langle x:y \rangle$	= parameter type x with meaning y
$\langle x y \rangle$	= exactly one of x or y
$\{ \dots \}$	= a comment (not part of the production)

SPACES are used for readability in the grammar description; SPACES in the actual metafile are indicated through the separator productions given below.

The metasymbols used in describing the grammar do not appear in the actual metafile.

4 Entering and Leaving the Metafile Environment

4.1 Generic Clear Text and Instantiations

The Clear Text Encoding is described in a generic fashion that permits it to be used with any character set capable of representing those characters enumerated in the Character Repertoire (see part 1, 4.7.6). An instantiation of the Clear Text Encoding is specified by defining the character set and coding technique to be used (for example, standard national character sets based on ISO 646, non-standard character sets such as EBCDIC, etc).

It is recommended that an instantiation of the Clear Text Encoding bound to the standard national character set based on ISO 646 be used in order to maximize portability of Clear Text metafiles between diverse systems. This also provides an encoding which can be incorporated into an ISO 2022 text environment as a complete code, to permit intermixing of text and graphics for applications which place a high priority on human readability.

4.2 Implicitly Entering the Metafile Environment.

The Clear Text 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, and if it is known by prior agreement which instantiation of the syntax is being used.

4.3 Invoking the CGM Coding Environment From ISO 2022.

For interchange with services using the code extension techniques of ISO 2022, the (standard national version) ISO 646 instantiation of the CGM Clear Text Encoding may be invoked from the 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 Code Extensions and substituted in this Standard prior to its publication.)

The first bit combination occurring after this escape sequence will then represent the beginning of a CGM metafile element or one of the "soft separators" or "null characters" defined below.

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 ISO 646 CGM coding environment with the ESC 2/5 F sequence. (The terms "designation" and "invocation" are defined in ISO 2022.)

It is permissible to make transitions between ISO 2022 and the metafile environment between pictures in the metafile as well as between metafiles. The state of the metafile interpreter and the state of the ISO 2022 environment are maintained separately and not stacked. The state of the metafile interpreter before BEGIN METAFILE or after END METAFILE is undefined, and sending a picture without a preceding BEGIN METAFILE and metafile descriptor is nonconforming interchange.

5 Metafile Format

A metafile in the Clear Text Encoding consists of a stream of characters forming a series of elements, each of which starts with an element name and ends with one of the element delimiters, either the SLASH character (also known as SLANT or SOLIDUS) or the SEMICOLON character. (Note: these characters do not act as element delimiters when occurring within the bounds of a string parameter, as defined below.)

5.1 Character Repertoire

In order to achieve objective (e) of sub-clause 0.2, the character repertoire of the Clear Text Encoding will be limited to those characters enumerated below, except for string parameters, which may contain any characters from the repertoire described in part 1 of this Standard (see part 1, 4.7.6).

- Upper-case characters:
 "A", "B", "C", "D", "E", "F", "G", "H", "I",
 "J", "K", "L", "M", "N", "O", "P", "Q", "R",
 "S", "T", "U", "V", "W", "X", "Y", "Z"
- Lower-case characters: (see note 1)
 "a", "b", "c", "d", "e", "f", "g", "h", "i",
 "j", "k", "l", "m", "n", "o", "p", "q", "r",
 "s", "t", "u", "v", "w", "x", "y", "z"
- Digits:
 "0", "1", "2", "3", "4", "5", "6", "7", "8", "9"
- " " (SPACE character)
- "+" (PLUS character)
- "-" (MINUS character)
- "#" (NUMBER SIGN)
- ";" (SEMICOLON character)
- "/" (SLASH, SLANT, or SOLIDUS character)
- "(" (LEFT or OPEN PARENTHESIS character)
- ")" (RIGHT or CLOSE PARENTHESIS character)
- "," (COMMA character)
- "." (DECIMAL POINT or PERIOD character)
- "'" (APOSTROPHE or SINGLE QUOTE character)
- "\"" (DOUBLE QUOTE character)
- "_" (UNDERSCORE character) (see note 2)
- "\$" (DOLLAR SIGN or CURRENCY symbol) (see note 2)
- "%" (PERCENT SIGN character)

NOTE 1: lower-case characters are considered to be the same as upper-case characters, when occurring outside of string parameters. Any combination of lower-case and upper-case characters may be used within an element or enumerated parameter name.

NOTE 2: The UNDERSCORE and DOLLAR SIGN symbols are defined as "null characters" within this encoding. They may appear anywhere within the metafile, and are mandated to have no effect on parsing (outside of string parameters). They are available for the generator or editor of the metafile to use in enhancing readability of tokens.

EXAMPLES: The following are all equivalent: linetype, LINETYPE, LineType, line_type, \$LINETYPE, L_I_N_E\$T_Y_P_E. The following are all equivalent: 123456, \$123456, 123_456, \$123_456, \$12\$34\$56.

Those control characters that are format effectors (BACKSPACE, CARRIAGE RETURN, LINEFEED, NEWLINE, HORIZONTAL TAB, VERTICAL TAB, and FORMFEED) are permitted in the metafile, but

Metafile Format

Character Repertoire

are treated as SPACE characters (that is, as soft delimiters) by the metafile interpreter whenever they occur outside of a string parameter. They may be used to assist in formatting the metafile to improve its readability. (The effect of such format effectors within string parameters is as defined in part 1 of this Standard.) A metafile written in the Clear Text Encoding is considered to be non-conforming interchange if it includes characters other than those listed in the repertoire and the format effectors (outside of string parameters). Implementation-dependent extensions which require use of characters other than the above should be embedded in the string parameters of the ESCAPE, MESSAGE, or APPLICATION DATA elements, or in comments.

Note that the code set of the characters is not fixed by this Standard. In order to accomplish the objective of editability, it is permitted to encode the Clear Text Encoding using the character set codes native to the system. It is presumed that standard conversion facilities can be used in translating Clear Text CGM files from one system's character set codes to another, consistent with the treatment of other text files being transferred between systems. It is recommended that the ISO 646 codes be used to encode Clear Text metafiles for transport between diverse systems.

Null characters or format effectors outside of text strings which do not exist in the target system's encoding may be dropped in such translation, and lower-case letters translated to upper case as necessary, without altering the information content of the metafile. Likewise, the two statement delimiter characters are interchangeable and may be changed in such a translation without affecting the information content of the metafile. The two string delimiter characters are interchangeable, but any translation shall correctly handle the possible occurrence of either string delimiter character within the string parameter.

5.2 Separators

5.2.1 Element Separators

<TERM> ::= <OPTSEP> <SLASH | SEMICOLON> <OPTSEP>

The SEMICOLON and SLASH characters may be used interchangeably to delimit elements in a Clear Text metafile. These elements do not, however, terminate an element when they occur within a string parameter, as described below.

The elements of the metafile are not terminated by the ends of records, as indicated by control characters such as CR (carriage return) or LF (linefeed). Multiple elements may exist on one line, and any element may extend over multiple lines.

5.2.2 Parameter Separators

The following productions are used in the Clear Text Encoding for parameter separators:

```
<SEPCHAR> ::= <SPACE | CARRIAGE RETURN | LINEFEED
              | HORIZONTAL TAB | VERTICAL TAB | FORMFEED>
<SOFTSEP>  ::= <SEPCHAR>+
<OPTSEP>   ::= <SEPCHAR>*
<HARDSEP>  ::= <OPTSEP> <COMMA> <OPTSEP>
<SEP>      ::= <SOFTSEP> | <HARDSEP>
```

Most commands require a SOFTSEP after the element name (e.g., at least one space). This permits element names to be formed from a mixture of alpha and numeric characters.

The separator between parameters is usually a SEP. This format permits omission of parameters. (Two consecutive COMMAS indicate an omitted parameter.)

Since the enclosing APOSTROPHE or DOUBLE QUOTE character sufficiently delineates string parameters, and the statement delimiter SLASH also sets off the data on either side of it, the separators between these characters and adjacent parameters or element names are optional (OPTSEP).

Separators

Metafile Format

SEPCHAR characters are not permitted within a name (element or enumerated type), or within the representation of a numeric parameter. Any place where a SEPCHAR is permitted (other than inside a string parameter), an arbitrary number of SEPCHARs may be used.

5.2.3 Comments in the Metafile

Comments may be included in a Clear Text metafile, to enhance its readability and usefulness. Some uses of comments might be to document hand-edited changes to the metafile, or as "notes to one's self" made while reading a metafile. To include other forms of nongraphical information in the metafile, it is suggested that the APPLICATION DATA element be used. If it is desired to convert a Clear Text metafile to one of the other encodings, comments may be either dropped or converted to APPLICATION DATA elements.

Comments are encoded as a series of printing characters and <SEPCHAR>s surrounded by "%" (PERCENT SIGN) characters. The text of the comment may not include this comment delimiter character.

Comments may be included any place that a separator may be used, and are equivalent to a <SOFT-SEP>; they may be replaced by a SPACE character in parsing, without affecting the meaning of the metafile.

5.3 Encoding of Parameter Types

5.3.1 Integer-Bound Types

INTEGERS, INTEGER COORDINATES, INDICES, and the components of COLOUR DIRECT parameters are all bound to signed integers, indicated in the encoding as I.

<I>	::=	<decimal integer> <based integer>
<decimal integer>	::=	<sign>o <digit>+
<sign>	::=	<PLUS SIGN> <MINUS SIGN>
<based integer>	::=	<sign>o <base> <NUMBER SIGN> <extended digit>+
<base>	::=	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
<digit>	::=	0 1 2 3 4 5 6 7 8 9
<extended digit>	::=	<digit> A B C D E F a b c d e f

The null characters are permitted within numbers, but are not shown in the productions for simplicity.

A decimal integer has an optional sign and at least one digit. If the sign appears, it immediately precedes the number with no intervening SPACE (or other <SEPCHAR>) characters allowed.

A based integer has an optional sign, a base (an unsigned integer in the range 2..16 inclusive, represented in base 10), a "#", and a string of one or more extended digits. If the sign appears, it immediately precedes the number with no intervening SPACE (or other <SEPCHAR>) characters allowed. The extended digits used shall be valid for the base named or the metafile is not conforming; e.g., for base 8 the digits "8", "9", etc. are not valid, for base 2 only the digits "0" and "1" are valid, and so forth. Case is not significant for the extended digits.

If the sign is omitted for either form, the number is considered non-negative.

Both the base and the <extended digit>+ are interpreted as unsigned numbers, and the final result negated if a MINUS SIGN preceded the number. No assumptions should be made as to the word size of

[ISO page 8]

Metafile Format

Encoding of Parameter Types

the metafile interpreter, or whether the underlying arithmetic is one's complement, two's complement, or sign-magnitude. For example, -1 would be encoded in hexadecimal as -16#1, -16#0001, etc. rather than 16#FFFF. Of course, metafiles may be created utilizing prior knowledge of the intended target machine, but any such assumptions will limit the portability of the metafile and are discouraged.

Examples:

0, 007, -5, +123_456

The following are equivalent: 65535, 16#FFFF, 16#ffff, 8#177777, 2#1111_1111_1111_1111

The following are equivalent: -32_768, -16#8000, -8#100000, -2#10000000_00000000

Interpretation of numerically bound parameters will be "free field", that is, there is an implied radix point to the right of the rightmost digit, and neither leading nor trailing spaces are significant. Leading zeroes are not significant.

5.3.2 Real-Bound Types

REALS and REAL COORDINATES are bound to real numbers, indicated in the encoding as R. These are written as either explicit-point numbers or scaled-real numbers (or decimal integers, where appropriate).

```

<R> ::= < explicit-point number > |
      < scaled-real number > |
      < decimal integer >

< explicit-point number > ::= <sign>o
                           <
                           <<digit>+ <PERIOD> <digit>*>
                           |
                           <<digit>* <PERIOD> <digit>+>
                           >

< scaled-real number > ::= <body> < E | e > <exponent>

<body> ::= <explicit-point number> |
          <decimal integer>

<exponent> ::= <decimal integer>

```

The interpretation of the scaled-real number is the same as standard scientific notation (similar to FORTRAN "E" format), where the number represented by <body> is multiplied by 10 taken to the power <exponent>.

There shall be at least one digit in an explicit-point number and in the body of a scaled-real number, which in the case of a single-digit number may appear on either side of the radix point. It is recommended but not required that there be at least one digit before the radix point, for numbers with only a fractional part. Zero may be encoded as "0.", ".0", "0.0", "0", etc., although the second form is not recommended.

In the case of a scaled-real number (one where an "E" or "e" appears), at least one digit shall appear in the <exponent>. No SPACE or other <SEPCHAR> characters are permitted between the <body> and the "E" or "e", or between the "E" or "e" and the <exponent>.

The interpretation of parameters bound to this data type will be "free field", that is, if there is an explicit radix point, it sets the radix point of the internal representation, and neither leading nor trailing spaces or zeroes are significant. If the radix point is omitted, it is implied to be at the right of the rightmost digit of the explicit-point number or of the <body> of the scaled-real number. Thus, decimal I-format

[ISO page 9]