

INTERNATIONAL STANDARD

ISO/IEC
8632-4

Second edition
1992-10-01

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

Part 4: Clear text encoding (standards.iteh.ai)

<https://standards.iteh.org/catalogue/standards-iso-17321-37-886-45c-iso-8632-4-1992>
*Technologies de l'information — Infographie — Métafichier de stockage
et de transfert des informations de description d'images —
Partie 4: Codage en clair des textes*



Reference number
ISO/IEC 8632-4:1992(E)

CONTENTS

1	Scope	1
2	Normative references	2
3	Notational conventions	3
4	Entering and leaving the metafile environment	4
4.1	Generic clear text and instantiations	4
4.2	Implicitly entering the metafile environment	4
4.3	Designating and invoking the CGM coding environment from ISO 2022	4
5	Metafile format	5
5.1	Character repertoire	5
5.2	Separators	6
5.2.1	Element separators	6
5.2.2	Parameter separators	6
5.2.3	Comments in the metafile	7
5.3	Encoding of parameter types	7
5.3.1	Integer-bound types	7
5.3.2	Real-bound types	8
5.3.3	String-bound types	9
5.3.4	Enumerated types	10
5.3.5	Derived types	10
5.3.6	Bitstream datatype	11
5.3.7	Structured data record operands	11
5.4	Forming names	11
5.4.1	Words deleted	11
5.4.2	Words added	12
5.4.3	Words used unabbreviated	12
5.4.4	Abbreviations	13
5.4.5	The derived element names	14

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/IEC 8632-4:1992
<https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-054097221d8/iso-iec-8632-4-1992>

6	Encoding the CGM elements	19
6.0.1	Comments on the element codings	19
6.0.2	Defaults and parameter range constraints	19
6.0.3	Allowable productions in clear text metafiles	19
6.1	Encoding delimiter elements	19
6.2	Encoding metafile descriptor elements	20
6.3	Encoding picture descriptor elements	28
6.4	Encoding control elements	31
6.5	Encoding graphical primitive elements	33
6.6	Encoding attribute elements	41
6.7	Encoding escape elements	48
6.8	Encoding external elements	49
6.9	Encoding segment control and segment attribute elements	49
7	Clear text encoding defaults	53
8	Clear text encoding conformance	54
A	Clear text encoding-dependent formal grammar	55
B	Clear text encoding example	56

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 8632-4:1992](https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992)

<https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992>

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 8632-4 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

This second edition cancels and replaces the first edition (ISO 8632-4:1987), 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. Annex B is for information only.

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 International Standards

The set of characters required to implement the Clear Text Encoding is a subset of those included in national versions of ISO/IEC 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 ISO/IEC 8632-1, 4.12.)

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 8632-4:1992](https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992)

<https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992>

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

Part 4 :

Clear text encoding

iTeh STANDARD PREVIEW (standards.iteh.ai)

1 Scope

This part of ISO/IEC 8632 specifies a clear text encoding of the Computer Graphics Metafile. For each of the elements specified in ISO/IEC 8632-1, 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.

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.

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*.

iTeh STANDARD PREVIEW (standards.iteh.ai)

[ISO/IEC 8632-4:1992](https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992)

<https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992>

3 Notational conventions

Unbracketed strings are terminals of this grammar. They appear in valid Clear Text data streams exactly as indicated in the specifications of this part, except for allowable variations on case and null characters described 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 <x:y> (see ISO/IEC 8632-1 for further explanation of these items).

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

<...>*	= star closure (0 or more occurrences).
<...>+	= plus closure (1 or more occurrences).
<...>0	= optional (exactly 0 or 1 occurrences).
<x:y>	= parameter type x with meaning y
<x y>	= exactly one of x or y
{...}	= a comment (not part of the production)
<...>(n)	= exactly n occurrences, n=0,1,2,...

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.

<https://standards.iteh.ai/catalog/standards/sist/173c6e37-af4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992>
 (standards.iteh.ai)

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, 4.7.3.2). 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/IEC 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/IEC 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 Designating and 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/IEC 646 instantiation of the CGM Clear Text Encoding may be designated and invoked from the ISO 2022 environment by the following escape sequence:

ESC 2/5 F <https://standards.iteh.ai/catalog/standards/sist/173c6e37-a4b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992>

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 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/IEC 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. 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 4.7.3.2 ISO/IEC 8632-1.

- 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:
"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)
- "\$" (DOLLAR SIGN or CURRENCY symbol)
- "%" (PERCENT SIGN character)

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.

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.

EXAMPLE — The following are all equivalent: `linetype`, `LINETYPE`, `LineType`, `line_type`, `SLINETYPE`, `L_I_N_E_S_T_Y_P_E`; similarly, 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

are treated as SPACE characters (that is, as soft delimiters) by the metafile interpreter whenever they occur outside of string parameters. 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 ISO/IEC 8632-1.

A metafile written in the Clear Text Encoding is considered to be non-conforming 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.

The code set of the characters is not fixed by this part of ISO/IEC 8632. 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/IEC 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

iTeh STANDARD PREVIEW (standards.iteh.ai)

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.

Metafile format**Separators**

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).

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 <SOFTSEP>; they may be replaced by a SPACE character in parsing, without affecting the meaning of the metafile.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

5.3 Encoding of parameter types**5.3.1 Integer-bound types**

Integers, integer coordinates, indexes, names, and the components of direct colour parameters are all bound to signed integers, indicated in the encoding as I.

The data types UI8 and UI32 of ISO/IEC 8632-1 are bound to non-negative values of signed integers, also indicated in this 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 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.

iTeh STANDARD PREVIEW (standards.iteh.ai)

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>.

Metafile format**Encoding of parameter types**

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 numbers are permitted to appear in a conforming metafile for parameters bound to real numbers when there is no fractional part.

For real numbers in all formats, the only permitted base of representation is base 10.

If the <sign> ("+" or "-") is omitted, the number is assumed to be non-negative. If the sign is present, it immediately precedes the body of the number, with no SPACE (or other <SEPCHAR>) characters allowed between it and the leftmost digit or radix point of the body of the number.

COMMA, SPACE and other <SEPCHAR> characters are not permitted within a number, but <NULLCHAR> characters are permitted (and have no effect on parsing).

Examples:

3.14159
7.853982E-7
271828e-5
42
-.04321 (not recommended form)
-0.043_21
42E2

iTeh STANDARD PREVIEW
(standards.iteh.ai)

[ISO/IEC 8632-4:1992](https://standards.iteh.ai/catalog/standards/sist/173c6e37-a44b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992)

<https://standards.iteh.ai/catalog/standards/sist/173c6e37-a44b-475a-aed3-05f4097221d8/iso-iec-8632-4-1992>

5.3.3 String-bound types

STRING parameters, both String (S) and String Fixed (SF) types, are represented as character strings immediately surrounded by a matched pair of either APOSTROPHE (SINGLE QUOTE) or DOUBLE QUOTE characters.

If an APOSTROPHE is required in a string delimited with APOSTROPHES, it is represented by two adjacent APOSTROPHES at that position in the string. Likewise, if a DOUBLE QUOTE character is required in a string delimited with DOUBLE QUOTE characters, it is represented by two adjacent DOUBLE QUOTE characters. For example, the following are equivalent

```
TEXT 0 0 FINAL "Murphy's Law: ""If it can go wrong, it will.""";
TEXT 0 0 FINAL 'Murphy's Law: "If it can go wrong, it will."';
```

DATA RECORD data type is represented as a string in this encoding.

STRING and DATA RECORD parameters are indicated in the element encodings as S. STRING FIXED is indicated in the element encodings as SF.