INTERNATIONAL STANDARD

ISO/IEC 9637-1

First edition
1994-03-01

Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Data stream binding —

Part 1: Character encoding

Technologies de l'information — Infographie — Techniques interfaciales de dialogues avec dispositifs graphiques (CGI) — Liaison de courant D —

Partie 1: Codage des caractères

Reference number
ISO/IEC 9637-1:1994(E)

Contents

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 9637-1 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Sub-Committee SC 24, *Computer graphics and image processing*, in collaboration with the European Computer Manufacturers Association (ECMA) and the European Conference of Postal and Telecommunications Administration (CEPT).

ISO/IEC 9637 consists of the following parts, under the general title *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Data stream binding*:

— *Part 1: Character encoding*

— *Part 2: Binary encoding*

— *Part 3: Clear text encoding*

Annexes A and B of this part of ISO/IEC 9637 are for information only.

Introduction

0.1 Purpose of the character encoding

The character encoding of the Computer Graphics Interface (CGI) provides a data stream representation of the CGI function syntax intended for situations in which it is important to minimize the size of the encoded data or transmit the data 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 9637-2 may be more appropriate.

0.2 Objectives

This encoding was designed with the following objectives:

a) *regular syntax:* All CGI functions are encoded in a uniform way so that parsing the encoded data is simple;

b) *compactness:* the encoding provides a highly compact data stream, suitable for systems with restricted storage capacity or transfer bandwidth;

c) *extensibility:* the encoding allows for future extensions;

d) *transportability:* the encoding is 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 Relationship to other International Standards

The character encoding has been developed in collaboration with the ISO/EIC JTC1/SC2. 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 9637 follows the rules of ISO/IEC 646 and ISO 2022.

For certain functions, ISO/IEC 9636 defines value ranges as being reserved for registration. The values and their meanings will be defined using the established registration procedures (see ISO/IEC 9636-1).

This encoding is based on that for the Computer Graphics Metafile ISO/IEC 8632-2. Wherever possible, the opcodes and parameter representations of CGM elements have been followed.

Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Data stream binding —

Part 1:
Character encoding

1 Scope

This part of ISO/IEC 9637 specifies a character encoding of the Computer Graphics Interface. For each of the functions specified in ISO/IEC 9636 an encoding is specified.

This encoding of the Computer Graphics Interface provides a highly compact representation of the data, suitable for applications that require the data 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 9637. At the time of publication the editions indicated were valid. All standards are subject to revision, and parties to agreements based to this part of ISO/IEC 9637 are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of ISO and IEC 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 — Coded extension techniques.*

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

ISO/IEC 6429:1992, *Information technology — Control functions for coded character sets.*

ISO 7942:1985, *Information processing systems — Computer graphics — Graphical Kernel System (GKS) functional description.*

ISO/IEC 8632-2:1992, *Information technology — Computer graphics — Metafile for the storage and transfer of picture description information — Part 2 : Character encoding.*

ISO/IEC 9636-1:1991, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 1: Overview, profiles and conformance.*

ISO/IEC 9636-2:1991, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 2: Control.*

ISO/IEC 9636-3:1991, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 3: Output.*

ISO/IEC 9636-4:1991, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 4: Segments.*

ISO/IEC 9636-5:1991, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 5: Input and echoing.*

ISO/IEC 9636-6:1991, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Functional specification — Part 6: Raster.*

ISO/IEC 9637-2:1992, *Information technology — Computer graphics — Interfacing techniques for dialogues with graphical devices (CGI) — Data stream binding — Part 2: Binary encoding.*

ISO/IEC TR 9973:1988, *Information processing — Procedures for Registration of Graphical Items.*

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

CEPT, *Rev. 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 9637 the bit combinations of a 7-bit code 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 9637 the bit combinations of an 8-bit code 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 octet 01000001, whereas 4/1 represents the 7-bit octet 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 CGI, these terms, and the concepts to which they refer, apply only within the string parameters of the functions listed in 6.9.4.1.

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 specified 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 CGI 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 CGI uses the G0 and G1/G2 sets within string parameters. The G3 set may be used within the string parameters of a conforming CGI data stream; this requires selection of the extended 7-bit or extended 8-bit mode in the CHARACTER CODING ANNOUNCER. ISO/IEC 9636 does not provide a function 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.

Bit b7 → b6 → b5 →	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b4 b3 b2 b1 / row \ col.	0	1	2	3	4	5	6	7
0 0 0 0	0	The C0 Set	2/0	A G-Set of 94 or 96 Bit Combinations				
0 0 0 1	1							
0 0 1 0	2							
0 0 1 1	3							
0 1 0 0	4							
0 1 0 1	5							
0 1 1 0	6							
0 1 1 1	7							
1 0 0 0	8							
1 0 0 1	9							
1 0 1 0	10							
1 0 1 1	11							
1 1 0 0	12							
1 1 0 1	13							
1 1 1 0	14							
1 1 1 1	15						7/15	

Figure 1 — The 7-bit code table.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	C0 Set		02/00	GL-Set of 94 or 96 Bit Combinations					C1 Set		10/00	GR-Set of 94 or 96 Bit Combinations				
01																
02																
03																
04																
05																
06																
07																
08																
09																
10																
11																
12																
13																
14																
15								07/15								15/15

Figure 2 — The 8-bit code table.

4 Overall structure

This encoding specifies representations for each of the CGI functions of ISO/IEC 9636 as well as any associated response. A function representation is the encoded representation of a function with its *In* parameters. A response representation is the encoded representation of the *Out* parameters of a soliciting function.

4.1 General form of the data stream

All function representations in the data stream are encoded using a uniform scheme. These are represented as variable length data structures, each consisting of opcode information designating the particular function representation, and the parameter data (if any).

All response representations in the return data stream are encoded using the same uniform scheme. These are represented as variable length data structures, each consisting of opcode information designating the particular response representation, and its return parameter data.

4.2 Entering and leaving the CGI environment

4.2.1 Implicitly entering the CGI environment

The CGI 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.

4.2.2 Designating and invoking the CGI coding environment from ISO 2022

For interchange with services using the code extension techniques of ISO 2022, the CGI 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 CGI function.

After the end of the CGI data (i.e., after the TERMINATE function) 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 CGI coding environment with the ESC 2/5 *F* sequence. (The terms *designation* and *invocation* are defined in ISO 2022.)

4.3 Encoding functions

The function descriptions in clause 5 and clause 6, as well as the formal grammars, of ISO/IEC

9636-2, ISO/IEC 9636-3, ISO/IEC 9636-4, ISO/IEC 9636-5, and ISO/IEC 9636-6, provide the basic syntax needed to encode each CGI function. They contain the function name and the input and output parameters, along with their abstract data types and value ranges. The order in which the parameters are specified in clause 5 and clause 6 is significant. This is the exact order in which they will be encoded in a representation.

The character encoding scheme has two basic components: the opcode and the parameter data. There may be no parameter data, a fixed amount of parameter data or an indefinite amount of parameter data depending on the function being represented.

CGI functions and their *In* parameters, if any, are encoded as function representations consisting of the opcode followed by any *In* parameter data. Functions with *Out* parameters are called soliciting functions. Their *Out* parameters are considered response data and are encoded in a separate response representation, consisting of an opcode followed by the *Out* parameter data (see 5.4).

The formats of the various function and response representations and the rules for their encoding in the character data stream are described below.

5 Method of encoding opcodes

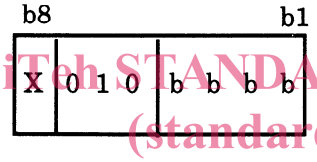
Each CGI function representation and response representation is composed of one opcode and parameters as required. The opcodes are coded as a sequence of bit combinations from columns 2 and 3 of the code chart. The encoding technique supplies:

- the basic opcode set;
- extension opcode sets.

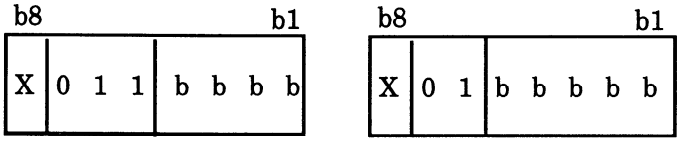
Parameter bytes are coded in columns 4 to 7 (with the exception of strings, which are otherwise delimited). This scheme permits an interpreter to skip the parameters of an function representation which it does not recognize or support. The interpreter can in such circumstances skip bytes until another opcode is encountered.

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 x 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* instances of 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* instances	<2/*x*>
2	<EOS>...<EOS> *n* instances	<3/*y* 2/*z*>
3	<EOS>...<EOS> *n* instances	<3/y 3/z>

<EOS> = 3/15
n = 0 selects the basic opcode set,
n = 1 selects the first extension opcode set,
n = *N* selects the *N-th* extension opcode set

x = 0,1,..,15
y = 0,1,..,14
z = 0,1,..,15

The number of opcodes supplied by this encoding technique (basic opcode set plus extension opcode sets) is 496*(*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

CGI opcodes overlap with opcodes from CGM and GKS. Where the same function, with identical parameters and identical semantics exists in two of them, then the same opcode (and parameterisation) is used.

Whereas all CGM and GKS opcodes are one and two-byte opcodes from the basic opcode set, some CGI opcodes extend into the first extension opcode set. These are three-byte opcodes, commencing with the byte 3/15.

5.4 Opcodes for soliciting functions

CGI functions which have *Out* (i.e. return) parameters have distinct opcodes in this encoding for the outward data, sent from generator to interpreter, and the return data, or 'response', returned from the interpreter.

There is a simple relationship between the outward opcode and the response opcode of a soliciting function. For all soliciting functions with an outward opcode *x*/*k a*/*b*, the response opcode will be of