INTERNATIONAL STANDARD

ISO/IEC 9637-2

First edition
1992-12-01

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

Part 2:

Binary encoding

Technologies de l'information — Infographie — Interfaces pour l'infographie — Interface du flux de données CGI —

Partie 2: Codage binaire

Reference number
ISO/IEC 9637-2:1992(E)

CONTENTS

Page

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 9637-2 was prepared by Joint Technical Committee ISO/IEC JTC1, *Information technology*.

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*

Annex A forms an integral part of this part of ISO/IEC 9637. Annex B is for information only.

Introduction

Purpose

The Binary Encoding of the Computer Graphics Interface (CGI), ISO/IEC 9636, provides a data stream representation of the CGI function syntax that can be optimized for speed of generation and interpretation, while still providing a standard means of interchange among computer systems. The encoding uses binary data formats that are more similar to the data representations used within computer systems than the data formats of the other encodings.

Some of the data formats may exactly match those of some computer systems. On most computer systems processing requirements for the Binary Encoding will be substantially lower than for the other encodings.

In cases where a computer system's architecture does not match the standard formats used in the Binary Encoding, and where absolute minimization of processing requirements is critical, and where interchange among dissimilar systems does not matter, it may be more appropriate to use a private encoding, conforming to the rules specified in ISO/IEC 9636-1.

Objectives

This encoding has the following features:

a) Partitioning of parameter lists: function/response representations are coded in the Binary Encoding by one or more partitions (see clause 4); the first (or only) partition of a representation contains the opcode (class code and id code);

b) Alignment of function representations and response representations: every function/response representation begins on a 16-bit boundary. Alignment of representations which follow partitions that require an odd number of 8-bit entities may require a partition to be padded with an 8-bit entity with all bits zero;

c) Uniformity of format: all function representations and response data records have an associated parameter length value. As a result, it is possible to ignore function representations which are not supported by the interpreter;

d) Efficiency of encoding parameter data: parameter data such as coordinates, indexes and colours are encoded as one or more 8-bit entities. The precision of every parameter is determined by the appropriate default precision or as set by a precision setting CGI function;

e) Extensibility: the arrangement of opcode class and id values has been designed to allow future growth;

f) Format of real data: real numbers are encoded using either IEEE floating point representation or a fixed-point representation;

g) Run length encoding option: if many adjacent colours have the same value, efficient encoding is possible. For each run a cell count is specified followed by the colour (or colour index);

h) Packed list encoding option: if adjacent colours do not have the same value, bit-stream lists are provided in which the values are packed as closely as possible;

i) Encoding of soliciting functions: the assignment of opcodes to functions which require a response has been designed so that all such functions can be recognized by a CGI interpreter;

j) Response Data: responses to soliciting functions have been assigned different opcodes from their associated soliciting functions. However, the response opcode can be derived in a straightforward manner from the soliciting function opcode;

k) Lists of data: there is a standard technique for representing lists of any type of data (with a few specific exceptions);

Relationship to other standards

This encoding is guided by the same objectives as the Computer Graphics Metafile Binary Encoding, ISO/IEC 8632-3:1992. For each CGI function which is identical in both semantics and parameterization to a CGM element, the encoding will be identical. That is, the opcodes will be identical and the parameters will use the same data type and appear in the same order. The extension mechanism defined in this encoding is also compatible with the CGM Binary Encoding.

The floating point representation of real data in this part of the Standard is that in ANSI/IEEE 754-1986.

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

For certain functions and response data, the CGI defines parameter value ranges as being reserved for registration. The values and their meanings will be defined using the procedures established in ISO TR 9973.

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

Part 2:
Binary encoding

1 Scope

This part of ISO/IEC 9637 specifies a Binary Encoding of the Computer Graphics Interface (CGI) data stream. For each of the function syntaxes in clause 5 and clause 6 of ISO/IEC 9636-2, ISO/IEC 9636-3, ISO/IEC 9636-4, ISO/IEC 9636-5, and ISO/IEC 9636-6, an encoding is specified in terms of an opcode and a sequence of parameters of specified data types. For each of these data types, an explicit representation in terms of bits, 8-bit and 16-bit entities is specified. For some data types, the exact representation depends on a type and/or precision for the data as used in the data stream.

The Binary Encoding of the Computer Graphics Interface data stream will, in many circumstances, reduce the effort required to generate and interpret the data stream as compared to other encodings.

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 revisions, and parties to agreements based on 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 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/IEC 7942:1985/Amd.1:1991, *Information processing systems – Computer graphics – Graphical Kernel System (GKS) functional description – Amendment 1*

ISO 8632-1:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information – Part 1: Functional specification.*

ISO 8632-3:1992, *Information technology – Computer graphics – Metafile for the storage and transfer of picture description information – Part 3: Binary 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 9637-1:1992, *Information technology – Computer graphics – Interfacing techniques for dialogues with graphical devices(CGI) – Data stream binding – Part 1: Character encoding.*

ISO/IEC TR 9973:1988, *Information technology – Computer graphics – Procedures for registration of graphical items.*

ANSI/IEEE 754, *Standard for Binary Floating Point Arithmetic.*

3 Definitions

3.1 representation: Portion of a binary-encoded function or response representation that contains the opcode (function/response class plus function/response id) and parameter length information. (See 4.3.4.)

3.2 octet: 8-bit entity in which all bits are significant. The bits are numbered from 7 (most significant) to 0 (least significant).

3.3 word: 16-bit entity in which all bits are significant. The bits are numbered from 15 (most significant) to 0 (least significant).

3.4 word-aligned: An entity is word-aligned when it begins on a word (16-bit) boundary within the data stream.

NOTE - Within this part of ISO/IEC 9637, the terms "octet", "word", and "word-aligned" have specific meanings. These meanings may not match those of a particular computer system on which this encoding of the data stream is used.

4 Overall structure

This encoding specifies representations for each of the CGI functions of ISO/IEC 9636 as well as any associated responses. 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 (function class plus function id) designating the particular function representation, the length of its parameter data and finally the parameter data itself (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 (response class plus response id) designating the particular response representation, the length of its return parameter data and finally the data itself.

4.2 General structure of the Binary Encoding

The octet is the fundamental unit of organization of the binary data stream. The Binary Encoding of the CGI data stream is a logical data structure consisting of a sequential collection of octets. Fields of two different sizes are defined within the Binary Encoding structure. These two sizes correspond to the octet, an 8-bit field, and the word, a 16-bit field. These fields are used in the remainder of this part of ISO/IEC 9637 for illustrating the contents and structure of function representations and parameters. The parameter list length of a function or response representation is expressed as a number of octets.

To optimize processing of the binary data stream on a wide collection of computers, data stream function and response representations are required to consist of an even number of octets. This forces the alignment of representations in the Binary Encoding data stream to word boundaries. It is necessary to pad a representation with null octets or bits to the word boundary if the parameter data does not fill to such a boundary. This padding of an extra octet or bits does not affect any parameter list length counts within a representation.

Parameter data can be organized into subgroups called partitions. Partitions are used to accommodate parameter data larger than that supported by the basic long-form representation, as described below. Partitions need not begin on a word boundary.

The bits of an octet are numbered 0 to 7, with 7 being the most significant. The bits of a word are numbered 0 to 15, with 15 being the most significant.

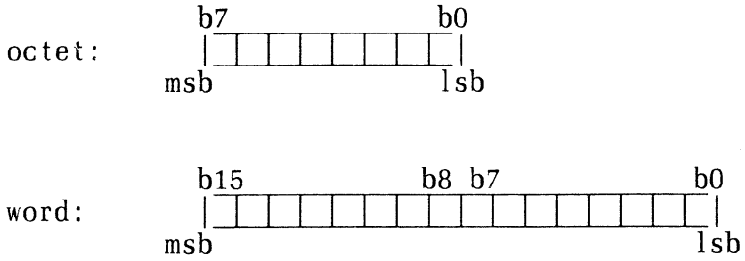

Figure 1 - Bit numbering for octets and words

If the consecutive bits of the binary data structure are numbered 1..*N*, the consecutive octets are numbered 1..*N*/8 (rounded to next integer), and the consecutive words are numbered 1..*N*/16 (rounded to next integer), then the logical correspondence of bits, octets, and words in the binary data structure is as illustrated in table 1.

Table 1 - Binary data structure

Data stream bit number	Octet bit number	Word bit number
1	b7/octet1	b15/word1
.	.	.
.	.	.
8	b0/octet1	b8/word1
9	b7/octet2	b7/word1
.	.	.
.	.	.
16	b0/octet2	b0/word1
17	b7/octet3	b15/word2
.	.	.
.	.	.
24	b0/octet3	b8/word2
25	b7/octet4	b7/word2
.	.	.
.	.	.

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 with very few exceptions. These special parameter encoding cases are specifically detailed. (See 6.1.)

The Binary Encoding scheme has two basic components: the representation header and the parameter data. The representation header provides information concerning the function opcode (function class and function id) and the amount (in octets) of parameter data that is being supplied in the function representation. There are one or more function representation headers for each function representation in the data stream. 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 one or more representation headers 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 one or more representation headers followed by the *Out* parameter data. (See 4.5.)

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

4.3.1 General structure of the representation header

Representations in the Binary Encoding have four forms - basic short-form representations, basic long-form representations, extended short-form representations and extended long-form representations. The forms differ in format with regard to the amount of parameter data accommodated and the number of function class and function id values accommodated. Specifically, the differences are:

-- the short-form representation always contains the complete function representation, including its parameter list data. The long-form representation is used to supply an indefinite amount of parameter data through the use of data partitions;

-- the short-form representation only accommodates parameter list data up to 30 octets in length. The long-form representation accommodates lengths up to 32767 octets per data partition with an indefinite number of partitions.

-- the basic form representations accommodate only 14 function classes and 127 function ids. The extended form representations accommodate an unlimited number of function classes and function ids.

The representation forms also differ in the format of the representation headers. There are specific fields, unique to each type of representation header, that identify the type of representation header. There are also differences in how the function opcode (function class and function id) and the parameter list length are encoded.

4.3.2 Basic short-form representation header

For the basic short-form, the representation header consists of a single word divided into three fields: function class, function id and parameter list length.

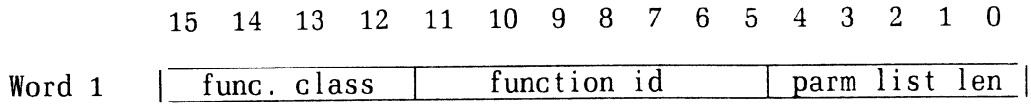

Figure 2 - Format of a basic short-form representation header.

The fields in the basic short-form representation header are as follows:

bits 15-12 function class (value range 0-14)

bits 11-5 function id (value range 0-127)

bits 4-0 parameter list length: the number of octets of parameter data that follow for this command (value range 0-30)

4.3.3 Basic long-form representation header

The representation header of the basic long-form representation consists of two words. The first word of the basic long-form representation has the same structure as the first word of the basic short-form representation with the difference that the parameter list length field contains the binary value 11111 (decimal 31). This value indicates that the header is a basic long-form representation header rather than a basic short-form representation header. The second word contains the length of the following parameter data partition and a flag that indicates if the parameter data partition is followed by another data partition or not.

An indefinite number of parameter partitions can be accommodated by the long-form representation. When the partition flag, bit 15 of the second word of the header has the value 1, it indicates that there will be another parameter data partition to follow. Each subsequent data partition of the function representation is preceded by a word composed of a partition flag and the partition's parameter list length in the same format as the second word of the representation header. The final data partition of a function representation is indicated by the partition flag being set to zero.

The parameter list length supplied in each of the parameter data partitions specifies the length of that partition in octets and not the length of the complete function representation data. Partitions need not start on a word boundary, therefore no padding octet needs to be appended to the data in a data partition which contains an odd number of octets for its parameter list length.

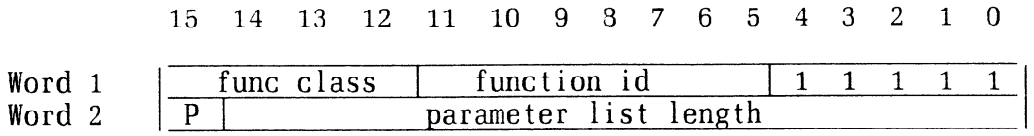

Figure 3 - Format of a basic long-form representation header.

The fields in the basic long-form representation header are as follows:

Word 1

bits 15-12 function class (value range 0-14)

bits 11-5 function representation id (value range 0-127)

bits 4-0 binary value 11111 (decimal 31) indicating long-form

Word 2

bit 15 P, partition flag
- 0 if 'final' partition
- 1 if 'not final' partition

bits 14-0 parameter list length: the number of octets of parameter data that follow for this partition (value range 0-32767).