Standard ECMA-35
6th Edition - December 1994

ECMA

Standardizing Information and Communication Systems

Character Code Structure and Extension Techniques

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **X.400:** C=ch; A=arcom, P=ecma, O=genevanet, OU1=ecma, S=helpdesk - **Internet:** Helpdesk@ECMA.CH

· PuTTY ·

Standard ECMA-35
6th Edition - December 1994

ECMA

Standardizing Information and Communication Systems

Character Code Structure and Extension Techniques

Phone: +41 22 849.60.00 - **Fax:** +41 22 849.60.01 - **X.400:** C=ch; A=arcom, P=ecma, O=genevanet, OU1=ecma, S=helpdesk - **Internet:** Helpdesk@ECMA.CH

IW-ECMA-035.DOC - 19.01.95 17,04

Brief History

ECMA/TC1 participates very actively in the work of JTC1/SC2 (previously ISO/TC97/SC2) on code structure and code extension, and contributed numerous technical papers to SC2/WG1, the group entrusted with the preparation of ISO 2022, the International Standard for code extension techniques. ECMA published its first Standard ECMA-35 on the same subject in 1971. Three further editions in 1980, 1982 and 1985 reflected the progress achieved internationally, and the text of the 1985 edition was identical with that of the 1986 edition of ISO 2022.

The present edition is technically almost identical with the 1985 edition of Standard ECMA-35 but is completely rearranged and rewritten to make it more convenient to use as a reference document. The present edition of Standard ECMA-35 is fully identical with the 1994 edition of ISO/IEC 2022.

Adopted as 6th edition of Standard ECMA-35 by the General Assembly of December 1994.

Table of contents

Section 1 - General

1 Scope

This Standard specifies the structure of 8-bit codes and 7-bit codes which provide for the coding of character sets. The code elements used in the structure are common to both the 8-bit and 7-bit codes. The codes use a variety of techniques for extending the capabilities of elementary 8-bit and 7-bit codes. Greater emphasis is given to 8-bit codes in this edition of the Standard than in previous editions because they are now more widely used.

The use of common elements in the 8-bit and 7-bit code structure enables any specific conforming 8-bit code to be transformed into an equivalent 7-bit code, and vice versa, in a simple and direct fashion.

ECMA-43 conforms to the 8-bit code structure specified here, and ECMA-6 conforms to the 7-bit code structure specified here.

NOTE 1

The coded character set specified in ISO/IEC 10646-1 has a different structure not in accordance with this Standard.

The code structure facilities specified here include various means of extending the number of control functions and graphic characters available in a code. They also include techniques to construct and formalize the definition of specific codes, and to provide a coded identification of the structure and of the constituent elements of such specific codes.

Specific codes may also be identified by means of object identifiers in accordance with ISO 8824, Abstract Syntax Notation One (ASN.1). The form of such object identifiers is specified in annex A.

Individual character sets and control functions intended for use with these 8-bit and 7-bit codes are assumed to be registered in the ISO International Register of Coded Character Sets to be Used with Escape Sequences, in accordance with ISO 2375 (see annex B). The register includes details to relate individual character sets and control functions with their coded representations, and also with the associated coded identifications of such character sets.

The principles established in this Standard may be utilized to form supplementary code structure facilities. For example ECMA-48 has followed such a procedure to formulate some parameterized control functions.

The use of uniform code structure techniques for the 8-bit and 7-bit codes specified here has the advantage of:

- permitting uniform provision for code structure in the design of information processing systems,
- providing standardized methods of calling into use agreed sets of characters,
- allowing the interchange of data between environments that utilise 8-bit and 7-bit codes respectively,
- reducing the risk of conflict between systems required to inter-operate.

When two systems with different levels of implementation of code structure facilities are required to communicate with one another, they may do so using the code structure facilities that they have in common.

The codes specified here are designed to be used for data that is processed sequentially in a forward direction. Use of these codes in strings of data which are processed in some other way, or which are included in data formatted for fixed-length record processing, may have undesirable results or may require additional special treatment to ensure correct interpretation.

NOTE 2

Since the fourth edition (1985) of this Standard the text has been completely rearranged and rewritten to make the Standard more convenient to use as a reference document. The fifth edition is arranged in three main sections as follows:

1 General
2 Character Sets and Codes
3 Code Identification and Escape Sequences

The sixth edition of this Standard is fully identical with International Standard ISO/IEC 2022:1994.

2 Conformance

2.1 Types of conformance

Full conformance to a standard means that all of its requirements are met. Conformance will only have a unique meaning if the standard contains no options. If there are options within the standard they must be clearly identified, and any claim of conformance must include a statement that identifies those options that have been adopted.

This Standard is of a different nature since it specifies a large number of facilities from which different selections may be made to suit individual applications. These selections are not identified in this Standard, but must be identified at the time that a claim of conformance is made. Conformance to such an identified selection is known as limited conformance.

The selection of facilities from this Standard that are to be used in a particular application will generally be included in a specification document, which states the adopted facilities and gives other details necessary to define fully one or more specific codes. Such a specification is said to be in accordance with this Standard (see 10.1).

2.2 Conformance of information interchange

A CC-data-element within coded information for interchange is in conformance with this Standard if the coded representations within that CC-data-element satisfy the following conditions:

a) they shall represent graphic characters, control functions, and code-identification functions in accordance with an identified selection of the facilities specified in this Standard (i.e. a version of this Standard, see 10.1);

b) when the code extension techniques specified in this Standard are used, they shall be implemented by the control functions and code-identification functions defined in this Standard with the meaning and coded representation specified in this Standard;

c) no coded representation that is either reserved for registration and not assigned, or reserved for future use, shall be used;

d) no registered escape sequence shall be used with a meaning different from that defined by the registration.

2.3 Conformance of devices

A device is in conformance with this Standard if it conforms to the requirements of 2.3.1, and either or both of 2.3.2 and 2.3.3 below. Any claim of conformance shall identify the document which contains the description specified in 2.3.1.

2.3.1 Device description

A device that conforms to this Standard shall be the subject of a description that

a) identifies either directly, or by reference to a specification that is in accordance with this Standard, the selection of facilities from this Standard that it can utilize when originating or when receiving CC-data-elements;

b) identifies the means by which the user may supply the corresponding characters and functions, or may recognize them when they are made available to the user, as specified in 2.3.2 and 2.3.3 respectively.

2.3.2 Originating devices

An originating device shall be capable of transmitting within a CC-data-element the coded representations of graphic characters from one or more graphic character sets, and of an identified selection of control functions and code-identification functions conforming to this Standard.

Such a device shall allow the user to supply, from an appropriate set, characters or other indications which will implicitly or explicitly determine the graphic characters, control functions, and code-identification functions whose coded representations are to be transmitted.

2.3.3 Receiving devices

A receiving device shall be capable of receiving within a CC-data-element and interpreting the coded representations of graphic characters from one or more graphic character sets, and an identified selection of control functions and code-identification functions conforming to this Standard.

Such a device shall make available to the user, from an appropriate set, characters or other indications which are implicitly or explicitly determined by the graphic characters, control functions, and code-identification functions whose coded representations are received.

3 References

ECMA-6:1991	7-bit Coded Character Set
ECMA-43:1991	8-bit Coded Character Set Structure and Rules
ECMA-48:1991	Control Functions for Coded Character Sets
ECMA-94:1986	8-bit Single-byte Coded Graphic Character Sets - Latin Alphabets No. 1 to No. 4
ISO 2375:1985	Data processing - Procedure for Registration of Escape Sequences
ISO 7498:1984	Information processing systems - Open Systems Interconnection - Basic Reference Model
ISO 8824:[1]	Information technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)
ISO 8825:[2]	Information technology - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)
ISO/IEC 10367:1991	Information technology - Standardized coded graphic character sets for use in 8-bit codes
ISO/IEC 10538:1991	Information technology - Control functions for text communication
ISO/IEC 10646-1:1993	Information technology - Universal Multiple-Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane

ISO International Register of Coded Character Sets to be Used with Escape Sequences

4 Definitions

For the purpose of this Standard, the following definitions apply.

4.1 bit combination

An ordered set of bits used for the representation of characters.

4.2 byte

A bit string that is operated upon as a unit.

NOTE 3

Each bit has the value either ZERO or ONE.

4.3 character

A member of a set of elements used for the organization, control or representation of data.

4.4 coded-character-data-element (CC-data-element)

An element of interchanged information that is specified to consist of a sequence of coded representations of characters, in accordance with one or more identified standards for coded character sets.

NOTE 4

In a communication environment in accordance with the Reference Model for Open Systems Interconnection of ISO 7498, a CC-data-element will form all or part of the information that corresponds to the Presentation-Protocol-Data-Unit (PPDU) defined in that Standard.

NOTE 5

[1] currently under revision

[2] currently under revision

When information interchange is accomplished by means of interchangeable media, a CC-data-element will form all or part of the information that corresponds to the user data, and not that recorded during formatting and initialization.

4.5 coded character set; code

A set of unambiguous rules that establishes a character set and the one-to-one relationship between the characters of the set and their bit combinations.

4.6 code extension

The techniques for the encoding of characters that are not included in the character set of a given code.

4.7 code table

A table showing the character allocated to each bit combination in a code.

4.8 combining character

A member of an identified subset of a coded character set, intended for combination with the preceding or following graphic character, or with a sequence of combining characters preceded or followed by a non-combining character.

4.9 control character

A control function the coded representation of which consists of a single bit combination.

4.10 control function

An action that affects the recording, processing, transmission or interpretation of data, and that has a coded representation consisting of one or more bit combinations.

4.11 to designate

To identify a set of characters that are to be represented, in some cases immediately and in others on the occurrence of a further control function, in a prescribed manner.

4.12 device

A component of information processing equipment which can transmit, and/or can receive, coded information within CC-data-elements.

NOTE 6

It may be an input/output device in the conventional sense, or a process such as an application program or a gateway function.

4.13 escape sequence

A string of bit combinations that is used for control purposes in code extension procedures. The first of these bit combinations represents the control function ESCAPE.

NOTE 7

In this Standard ESCAPE is always referred to as a control character.

4.14 Final Byte

The bit combination that terminates an escape sequence or a control sequence.

4.15 graphic character

A character, other than a control function, that has a visual representation normally handwritten, printed or displayed, and that has a coded representation consisting of one or more bit combinations.

4.16 graphic symbol

A visual representation of a graphic character or of a control function.

4.17 Intermediate Byte

A bit combination which may occur between that of the control character ESCAPE and the Final Byte in an escape sequence.

4.18 to invoke

To cause a designated set of characters to be represented by the prescribed bit combinations whenever those bit combinations occur.

4.19 repertoire

A specified set of characters that are each represented by one or more bit combinations of a coded character set.

4.20 to represent

a) To use a prescribed bit combination with the meaning of a character in a set of characters that has been designated and invoked; or

b) To use an escape sequence with the meaning of an additional control function.

4.21 user

A person or other entity that invokes the services provided by a device.

NOTE 8

This entity may be a process such as an application program if the "device" is a code convertor or a gateway function, for example.

NOTE 9

The characters, as supplied by the user or made available to the user, may be in the form of codes local to the device, or of non-conventional visible representations, provided that 2.3 above is satisfied.

5 Notation, code tables and names

5.1 Notation

The bits of the bit combinations of the 8-bit code are identified by b_8, b_7, b_6, b_5, b_4, b_3, b_2 and b_1, where b_8 is the highest order, or most-significant, bit and b_1 is the lowest-order, or least-significant, bit.

The bits of the bit combinations of the 7-bit code are identified by b_7, b_6, b_5, b_4, b_3, b_2 and b_1, where b_7 is the highest order, or most-significant, bit and b_1 is the lowest-order, or least-significant, bit.

The bit combinations may be interpreted to represent integers in binary notation, in the range 0 to 255 for the 8-bit code, and in the range 0 to 127 for the 7-bit code, by attributing the following weights to the individual bits:

Bit:	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Weight:	128	64	32	16	8	4	2	1

In this Standard, the bit combinations are identified by notations of the form x/y, where x and y are numbers in the range 00 to 15.

The correspondence between the notations of the form x/y and the bit combinations consisting of the bits b_8 or b_7 to b_1 is as follows:

– x for the 8-bit code is the number represented by b8, b7, b6, and b5 where these bits are given the weights 8, 4, 2 and 1 respectively;

– x for the 7-bit code 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 b_4, b_3, b_2 and b_1 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 the row number (see 5.2).

5.2 Code tables

An 8-bit code table consists of 256 positions arranged in 16 columns and 16 rows. The columns and rows are numbered 00 to 15 (see figure 1).

A 7-bit code table consists of 128 positions arranged in 8 columns and 16 rows. The columns are numbered 00 to 07 and the rows 00 to 15 (see figure 1).

The code table positions are identified by notations of the form x/y, where x is the column number and y is the row number. By convention, leading zeroes are included in the column and row numbers (e.g. 02/01).

The positions of the code table are in one-to-one correspondence with the bit combinations of the code. The notation of a code table position, of the form x/y, is the same as that of the corresponding bit combination.

	00	01	02	03	04	05	06	07
00								
01								
02								
03								
04								
05								
06								
07								
08								
09								
10								
11								
12								
13								
14								
15								

7-bit

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00																
01																
02																
03																
04																
05																
06																
07																
08																
09																
10																
11																
12																
13																
14																
15																

8-bit

Figure 1 - Code tables

5.3 Names of characters

This Standard assigns one name to each character. In addition, it specifies an acronym for each control character and for the characters SPACE and DELETE. By convention, only capital letters, space and hyphen are used for writing the names of the characters. For acronyms only capital letters and digits are used. It is intended that the acronyms and this convention be retained in all translations of the text.

Section 2 - Character sets and codes

6 Characters and character sets

6.1 Types of characters and character sets

The structure of 8-bit and 7-bit codes specified by this Standard makes use of the following types of characters, character sets, and functions:

- fixed coded characters,
- sets of coded graphic characters,
- sets of coded control functions (or control characters),
- coded single additional control functions.

These components are specified respectively in 6.2 to 6.5 below.

The coded representations of the graphic characters and control functions are specified in relation to the 8-bit and 7-bit code tables defined in 5.2 above. A coded representation for each type of component is specified within columns 00 to 07 of the 8-bit and 7-bit code tables. For some components an alternative coded representation is specified in columns 08 to 15 of the 8-bit code table, and is not applicable to any 7-bit code.

6.2 Fixed coded characters

6.2.1 Character DELETE

Name: DELETE Acronym: DEL Coded representation: 07/15

DEL was originally used to erase or obliterate an erroneous or unwanted character in punched tape. DEL may be used for media-fill or time-fill. DEL characters may be inserted into, or removed from, a CC-data-element without affecting its information content, but such action may affect the information layout and/or the control of equipment.

6.2.2 Character ESCAPE

Name: ESCAPE Acronym: ESC Coded representation: 01/11

ESCAPE is a control character used for code extension purposes. It causes the meaning of a limited number of the bit combinations following it in a CC-data-element to be changed. These bit combinations, together with the preceding bit combination that represents the ESC character, constitute an escape sequence.

Escape sequences provide the coded representations of code-identification functions and of some types of control functions. The various uses of escape sequences are specified in clause 13. Code identification functions are specified in clauses 14 and 15.

6.2.3 Character SPACE

Name: SPACE Acronym: SP Coded representation: 02/00

SPACE is a graphic character. It has a visual representation consisting of the absence of a graphic symbol. It causes the active position to be advanced by one character position.

6.3 Sets of coded graphic characters

6.3.1 Types of coded graphic character set

A graphic character shall have a coded representation comprising one or more 8-bit combinations (bytes) in an 8-bit code, and one or more 7-bit combinations (bytes) in a 7-bit code. Within a coded graphic character set each character shall be represented by the same number of such bit combinations.

The bit combinations used to represent the graphic characters in a set shall be either from the six adjacent columns numbered 02 to 07 of the code tables or from the six adjacent columns numbered 10 to 15 of the 8-bit code table.

The type of a coded graphic character set is defined by the maximum number of graphic characters that the set can contain. The types of set specified here are illustrated in figure 3.

A coded graphic character set in which each character is represented by a single bit combination shall be one of the following:

– 94-character set, in positions 02/01 to 07/14, or 10/01 to 15/14;
(i.e. all positions in columns 02 to 07 except 02/00 and 07/15, or
all positions in columns 10 to 15 except 10/00 and 15/15)

– 96-character set, in positions 02/00 to 07/15, or 10/00 to 15/15.
(i.e. all positions in columns 02 to 07, or in columns 10 to 15)

In a 94-character set no character shall be allocated to positions 02/00 and 07/15.

A coded graphic character set in which each character is represented by a sequence of n bit combinations, where n>1, shall be one of the following:

– 94^n-character set,
– 96^n-character set.

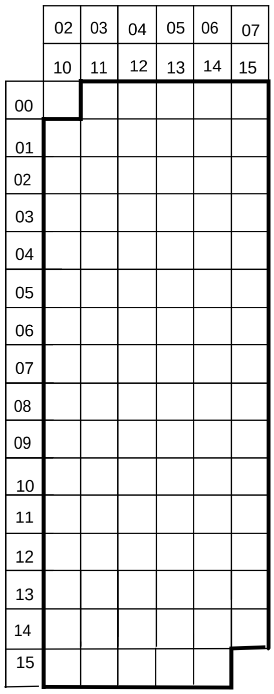
These sets are here referred to as multiple-byte sets.

A 94^n-character set shall consist of up to 94^n graphic characters each of which is represented by a sequence of n 8-bit or 7-bit combinations, either all in the range 02/01 to 07/14 or all in the range 10/01 to 15/14. In a 94^n-character set no character shall have a coded representation that includes the bit combination 02/00 or 07/15.

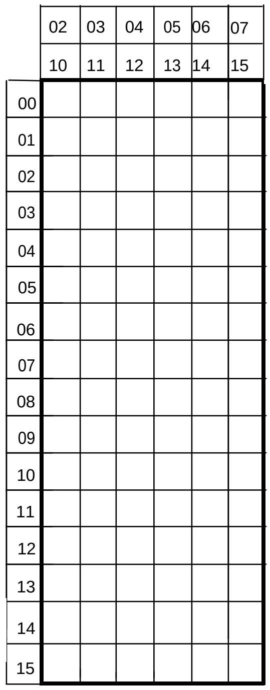
A 96^n-character set shall consist of up to 96^n graphic characters each of which is represented by a sequence of n 8-bit or 7-bit combinations, either all in the range 02/00 to 07/15 or all in the range 10/00 to 15/15.

NOTE 10

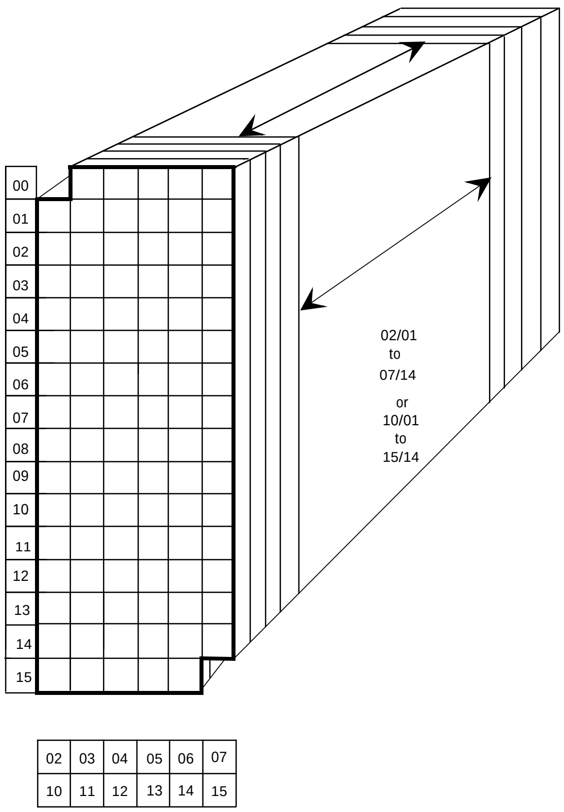
The 8th bit (b_8) of each byte in such an 8-bit multiple-byte representation is uniformly either ZERO or ONE.



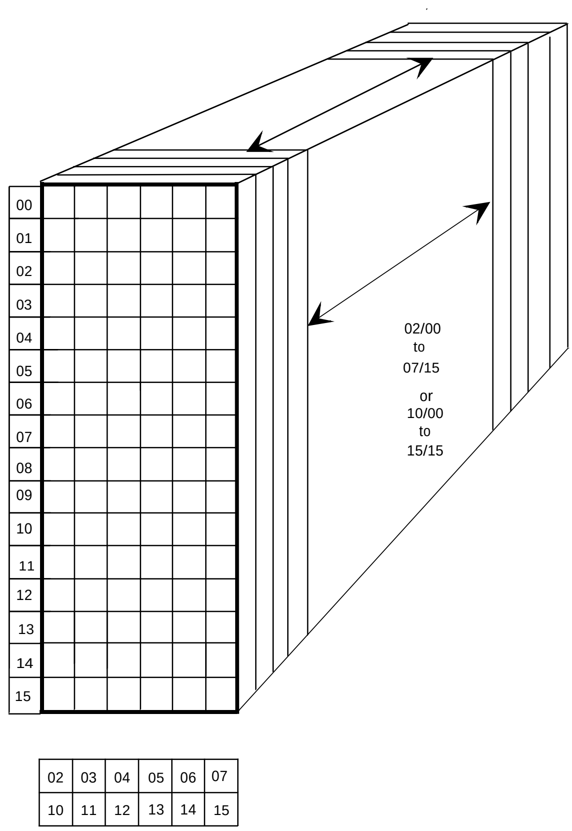
94-character set



96-character set



94 x 94-character set



96 x 96-character set

94-0112-A

Figure 2 - Structure of sets of coded graphic characters

6.3.2 Contents of a coded graphic character set

Within a coded graphic character set either a unique graphic character shall be allocated to each of the (sequences of) bit combinations that are specified for that set, or that bit combination (or sequence) shall be declared unused.

Any coded graphic character set shall not contain the characters SPACE or DELETE, or any control character (see 6.4). However, characters other than SPACE and representing spaces of different sizes or usage may be assigned to any (sequences of) bit combinations in any set of graphic characters.

6.3.3 Combination of graphic characters

Unless specifically defined otherwise, graphic characters shall not be combining characters, i.e. they shall not be intended for combination with an adjacent graphic character.

Some graphic character sets may allow for the graphical representation of additional graphic symbols, such as accented letters, by the imaging of two or more graphic characters as a single graphic symbol. Two combination methods are recognised in this Standard:

a) graphic characters that are non-combining characters may be combined by the use of the control character BACKSPACE or CARRIAGE RETURN;

b) graphic characters that are specified to be combining characters may be used in conjunction with a non-combining graphic character.

Sponsors of graphic character sets who apply for registration according to ISO 2375 are expected to identify any combining characters that are in the set.

NOTE 11

A standard that defines a character set should specify which characters, if any, are combining characters, and how they may be used, since a registration does not require such details to be stated.

NOTE 12

The graphic character set of ECMA-6 allows for the first of the above methods for the imaging of accented characters.

NOTE 13

ECMA-48 specifies a third method for combining graphic characters, independent of the specification of the characters themselves, by the use of the control function GRAPHIC CHARACTER COMBINATION (GCC).

6.3.4 Sources of coded graphic character sets

Sets of graphic characters and their coded representations are specified in other standards such as ECMA-6 or ISO/IEC 10367, and in national standards. Some of these sets, and some additional sets, are specified in the ISO International Register of Coded Character Sets (see annex B).

NOTE 14

New and revised character sets may be added to the register when required.

Sets of graphic characters for private use may be defined by agreement between the interchange parties.

6.4 Sets of coded control functions

6.4.1 Types of coded control function set

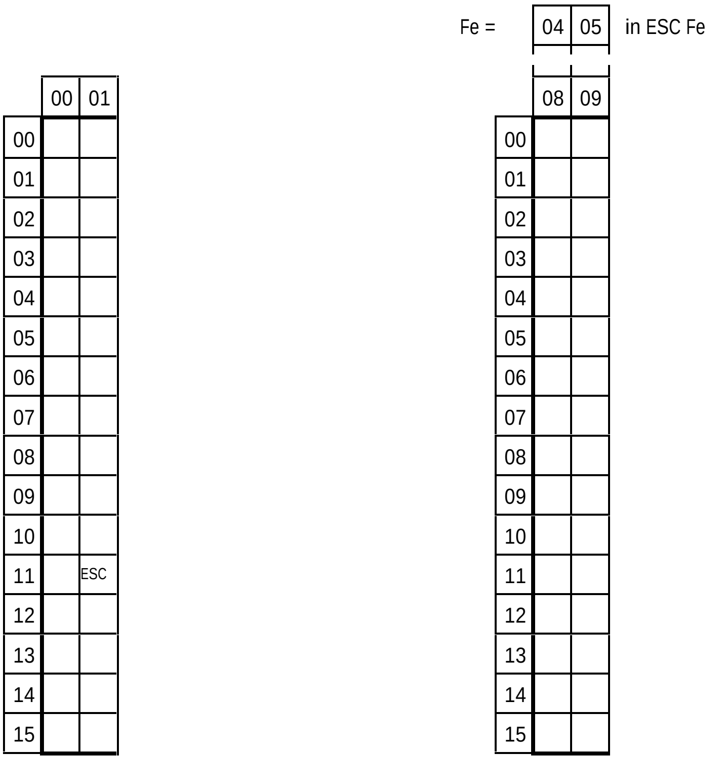
A set of coded control functions shall contain up to 32 control functions (or control characters) allocated to two adjacent columns of a code table.

Two types of coded control function set are defined as follows:

- primary set, in positions 00/00 to 01/15,
- supplementary set, in positions 08/00 to 09/15, or represented by escape sequences.

A primary set shall include the ESCAPE character. A supplementary set shall not include that character. These sets are illustrated in figure 3.

Either a unique control function shall be allocated to each position or the position shall be declared unused.



Primary set   Supplementary set

Figure 3 - Structure of sets of coded control functions (or characters)

6.4.2 Primary sets of coded control functions

A control function in a primary set shall have a coded representation consisting of one 8-bit or 7-bit combination, i.e. it is a control character.

A primary set of coded control functions shall include the control character ESCAPE in position 01/11.

If any control function from the primary set specified in ECMA-48 is included, it shall have the definition and the coded representation specified therein. No transmission control characters, other than the ten specified in ECMA-48, shall be included in a primary set of coded control functions.

6.4.3 Supplementary sets of coded control functions

A control function in a supplementary set shall have a coded representation consisting of one 8-bit or 7-bit combination when the set is invoked in positions 08/00 to 09/15. It shall be represented by an escape sequence of type Fe (see 13.2) otherwise.

NOTE 15

The notation Fe indicates a bit combination in the range 04/00 to 05/15. The escape sequence consists of the two bit combinations ESC Fe (13.2.5).

A supplementary set of coded control functions shall not include the control character ESCAPE or any of the transmission control functions of the primary set of ECMA-48.

6.4.4 Sources of coded control function sets

Control functions for a wide variety of applications are specified in ECMA-48. A standardized primary set and supplementary set are included (identified there as C0 and C1 sets). Sets of control functions are also registered in the ISO International Register of Coded Character Sets (see annex B). Each set is registered either as a primary (C0) set only, or as a supplementary (C1) set only.

NOTE 16

New and revised sets of coded control functions may be added to the register when required.

Sets of coded control functions for private use may be defined by agreement between the interchange parties.

6.5 Coded single additional control functions

A coded single additional control function shall be either:

- a standardized single control function, or
- a registered single control function, or
- a private control function.

Each such function shall be represented by an escape sequence (see clause 13).

6.5.1 Standardized single control functions

A standardized single control function shall have a permanently assigned meaning. Such a function shall be represented by an escape sequence of type Fs (13.2.1). Each such function shall be registered, together with its coded representation, in the ISO International Register of Coded Character Sets (see annex B).

NOTE 17

Any candidates for registration as standardized control functions must first be approved by ISO/IEC JTC1/SC2. If approval is granted the control function is registered according to the procedure of ISO 2375. It will normally then be specified in a standard published by ISO or other recognised body.

NOTE 18

The notation Fs indicates a bit combination in the range 06/00 to 07/14. The escape sequence consists of the bit combinations ESC Fs (13.2.5).

6.5.2 Registered single control functions

A registered single control function shall have a permanently assigned meaning. Such a function shall be represented by an escape sequence of type 3Ft (13.2.2). Each such function shall be registered, together with its coded representation, in the ISO International Register of Coded Character Sets (see annex B).

NOTE 19

The notation Ft indicates a bit combination in the range 04/00 to 07/14. The escape sequence consists of the bit combinations ESC 02/03 .. Ft (13.2.5).

6.5.3 Private control functions

Private control functions have no standardized meaning. They are for private use and may be defined by agreement between the interchange parties. A private control function shall be represented by an escape sequence of type Fp or of type 3Fp (13.2.2).

NOTE 20

The notation Fp indicates a bit combination in the range 03/00 to 03/15. The escape sequences consist respectively of the bit combinations ESC Fp and ESC 02/03 .. Fp (13.2.5).

6.5.4 Sources of coded single control functions

Some standardised single control functions are specified elsewhere in this Standard, see 7.3 and 15.3, and some are specified in ECMA-48.

Registered control functions are found in the ISO International Register of Coded Character Sets (see annex B).

Private control functions are defined by agreement between the interchange parties.

7 The elements of 8-bit and 7-bit codes

7.1 Summary of the elements

An element of an 8-bit or a 7-bit code shall be either:

- a coded character-set (7.2),
- a coded single additional control function (6.5),
- a coded code-identification function (7.4).

These code elements are illustrated in figure 4.

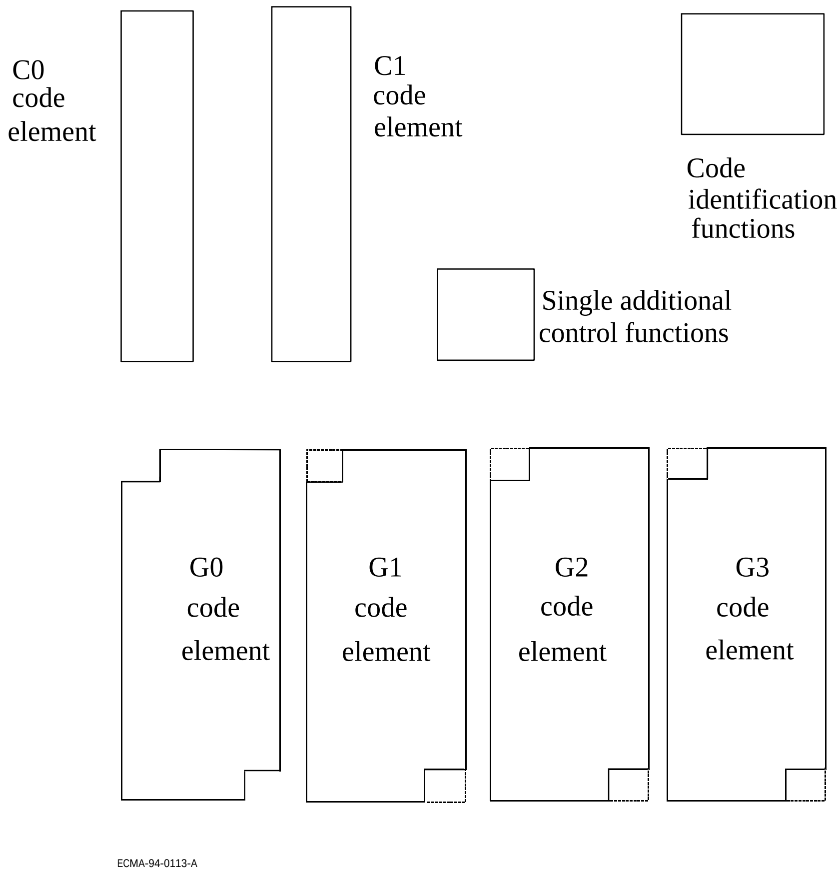


Figure 4 - Elements of a code

7.2 Character-set code elements

A character-set code element shall be an identified set of coded graphic characters, or of coded control functions (or characters), together with an element name to indicate the relationship of the set to the structure of the code. When the element is invoked, the corresponding set shall be represented in those columns of an 8-bit or 7-bit code table that are specified in 6.3.1, 6.4.2, or 6.4.3 for that type of set.

A character-set code element shall be one of those shown in table 1 below. The table shows the name of the element, the type of coded character set that it comprises, and the column numbers of the 8-bit or 7-bit code tables into which it may be invoked.

Table 1 - Character-set code elements

Name	Column numbers	Type of coded character set
C0	00 and 01	Control functions (characters), primary set
C1	08 and 09 or ESC Fe	Control functions, supplementary set
G0	02 to 07	Graphic characters - 94-character or 94^n-character set
G1	02 to 07 or 10 to 15	Graphic characters - 94-character or 94^n-character or 96-character or 96^n-character set
G2	(as for G1)	(as for G1)
G3	(as for G1)	(as for G1)

NOTE 21

The identification of specific graphic character sets as the elements G0, G1, G2, and G3, and the identification of specific control function sets as the elements C0 and C1, is referred to in this Standard by the term "designation". Designation of sets may be achieved by the use of designation functions (7.4) or by other methods (see 10.2).

7.3 Invocation of character-set code elements

The designation of a control character set as a C0 or C1 code element shall invoke that set.

The designation of a graphic character set as a G0, G1, G2, or G3 code element shall invoke that set if the code element already has a shift status (8.3.3 and 9.3.3); otherwise the use of a corresponding shift function shall invoke that set. Shift functions are control functions, and are specified in 8.3, 8.4, 9.3, and 9.4. They are listed in table 2 below.

Table 2 shows the name, acronym, and coded representation of each shift function. The entry in the "usage code" column signifies whether the function is available for use in an 8-bit code or a 7-bit code as follows:

– 7 7-bit code only,
– 8 8-bit code only,
– 7/8 7-bit and 8-bit codes.

The entry in the "type" column signifies the allocation of the function to a particular code element as follows:

– C0 a member of the primary set of control functions,
– C1 a member of the supplementary set of control functions,
– Fs a standardised single control function.

Table 2 - Shift functions

Name	Acronym	Usage Code	Coded Representation	
			Type	Bit Combination
SHIFT-IN	SI	7	C0	00/15
SHIFT-OUT	SO	7	C0	00/14
LOCKING-SHIFT ZERO	LS0	8	C0	00/15
LOCKING-SHIFT ONE	LS1	8	C0	00/14
LOCKING-SHIFT TWO	LS2	7/8	Fs	ESC 06/14
LOCKING-SHIFT THREE	LS3	7/8	Fs	ESC 06/15
SINGLE-SHIFT TWO	SS2	7/8	C1	ESC 04/14 or 08/14
SINGLE-SHIFT THREE	SS3	7/8	C1	ESC 04/15 or 08/15
LOCKING-SHIFT ONE RIGHT	LS1R	8	Fs	ESC 07/14
LOCKING-SHIFT TWO RIGHT	LS2R	8	Fs	ESC 07/13
LOCKING-SHIFT THREE RIGHT	LS3R	8	Fs	ESC 07/12

NOTE 22

The coded representations of LS2, LS3, SS2, SS3, LS1R, LS2R, and LS3R, are allocated in the ISO International Register of Coded Character Sets (see annex B), and are repeated here for convenience.

NOTE 23

If a 7-bit single-byte representation of SS2 and SS3 is required, it should be bit combination 01/09 and 01/13, respectively in the primary set of control functions (see annex B of ISO/IEC 10538).

When any shift function from table 2 is required for use in an 8-bit or 7-bit code it shall be included in, or as, the appropriate element of that code, in accordance with the "type" entry above.

7.4 Coded code-identification functions

The following types of coded code-identification functions are specified in this Standard:

- designation of sets of control characters (14.2),
- designation of sets of graphic characters (14.3),
- identify revision number of character sets (14.5),
- announcement of code structure and facilities (15.2),
- code switching (15.4).

An associated control function is also specified:

- data delimiter (15.3).

These functions may be included as code elements in a 8-bit or 7-bit code when required. Alternative methods of providing equivalent facilities may be specified in standards for information interchange (see 10.2).

7.5 Unique coding of graphic characters

The same character may be present in more than one of the sets of graphic characters that have been designated as the G0, G1, G2, and G3 code elements of an 8-bit or 7-bit code. Such a character shall be regarded as the same character as a character in another of those code elements if both characters have the same name within the specifications, or entries of the ISO International Register of Coded Character Sets, that respectively define the two sets.

If the same character has been allocated to more than one of those sets, then that character may be represented by the coded representation taken from any of the code elements G0, G1, G2, or G3 in which the character has been allocated.

Where a particular application of this Standard requires that every character in information interchange should have a unique coded representation, the specification of the version of the code (10.1) shall state that restriction.

When the restriction for unique coding applies, any character shall be represented by the coded representation taken from the lowest numbered code element (in the sequence G0, G1, G2, G3) in which the character has been allocated. In this case a coded representation for such a character within one of the other, higher numbered code elements shall not be used, even if the higher numbered code element is already invoked and the lowest numbered code element in which the character is allocated is not currently invoked.

8 Structure of 8-bit codes

8.1 Code table layout for 8-bit codes

An 8-bit code shall have a structure which is based on an 8-bit code table arranged in separate areas as follows (see figure 5):

- columns 00 and 01 shall be the CL area,
- columns 02 to 07 shall be the GL area,
- columns 08 and 09 shall be the CR area,
- columns 10 to 15 shall be the GR area.

The bit combinations in these areas shall be used to represent characters as follows:

- CL, a primary set of control characters;

- GL, either the characters SPACE (6.2.3) and DELETE (6.2.1) and a set of 94 or 94^n graphic characters, or a set of 96 or 96^n graphic characters;
- CR, either a supplementary set of control functions, or unused;
- GR, either a set of 94 or 94^n graphic characters (leaving positions 10/00 and 15/15 unused), or a set of 96 or 96^n graphic characters, or unused.

NOTE 24

In figure 5, positions 02/00 and 07/15, or positions 10/00 and 15/15, may be used to represent any graphic characters when a 96-character or 96^n-character set is invoked in the GL or GR area respectively.

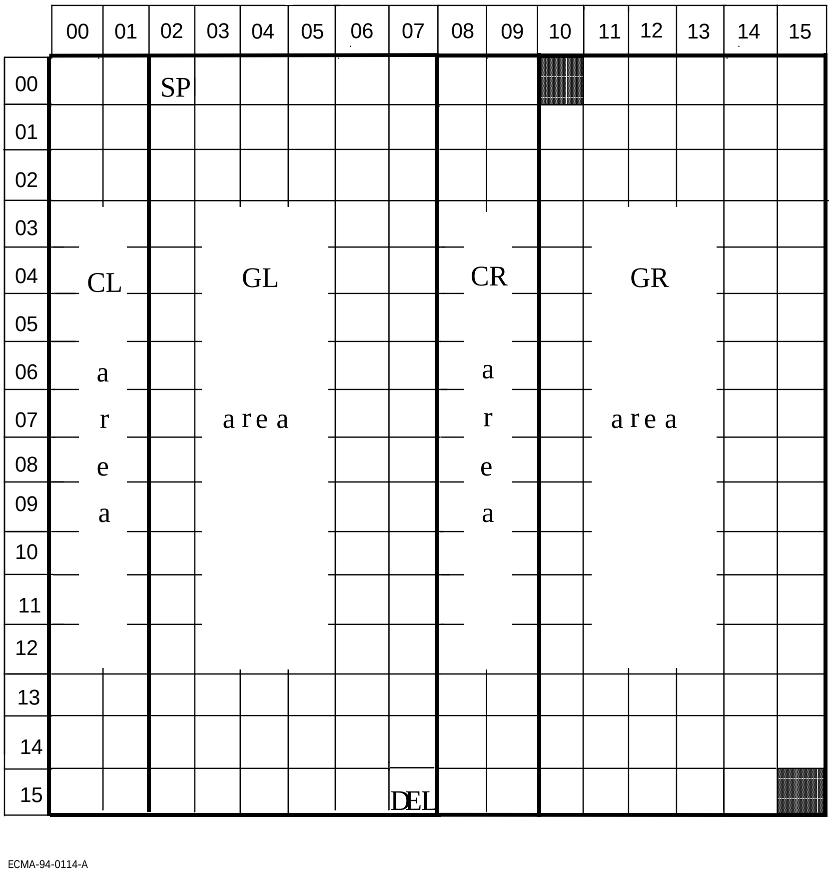


Figure 5 - Areas of 8-bit code table

8.2 Elements and structure of the code

An 8-bit code shall include a C0 and a G0 code element, and may also include some or all of the code elements C1, G1, G2, and G3 (7.2). If more than two of G0 to G3 are included, it shall include appropriate shift functions (7.3) to enable the graphic characters in those sets to be invoked.

An 8-bit code may include one or more single additional control functions (6.5), or code-identification functions (7.4).

If more than four sets of graphic characters, or more than two sets of control functions, are required in a specific 8-bit code, designation functions may be included within that code to permit other character sets to be designated as code elements G0 to G3, C0, or C1 (14.1).

The structure of the code is illustrated in figure 6.

8.3 Invocation of graphic character sets by means of shift functions

8.3.1 LOCKING-SHIFT ZERO, .. ONE, .. TWO, and .. THREE

Each of the four locking-shift functions LS0, LS1, LS2, and LS3 (7.3) shall invoke in the GL area the specific graphic character set that has been identified as the respective code element G0, G1, G2, or G3. The graphic character set that is invoked shall be the set that was last designated as the respective element G0, G1, G2, or G3 of the code. When such a set has been invoked the characters in that set shall be represented by the corresponding bit combinations (or sequences of n bit combinations for a multiple-byte set) in the GL area of the code table.

When a locking-shift function LS0, LS1, LS2, or LS3 invokes a 94-character or 94^n-character set, that set shall be invoked into positions 02/01 to 07/14, SPACE shall be invoked into position 02/00, and DELETE shall be invoked into position 07/15. When a locking-shift function LS1, LS2, or LS3 invokes a 96-character or 96^n-character set, that set shall be invoked into positions 02/00 to 07/15. The characters SPACE and DELETE cannot be represented while such a set is in an invoked state.

8.3.2 LOCKING SHIFT ONE RIGHT, .. TWO RIGHT , and .. THREE RIGHT

Each of the three locking-shift functions LS1R, LS2R, and LS3R (7.3) shall invoke in the GR area the specific graphic character set that has been identified as the respective code element G1, G2, or G3. The graphic character set that is invoked shall be the set that was last designated as the respective element G1, G2, or G3 of the code. When such a set has been invoked the characters in that set shall be represented by the corresponding bit combinations (or sequences of n bit combinations for a multiple-byte set) in the GR area of the code table.

When a locking-shift function LS1R, LS2R, or LS3R invokes a 94-character or a 94^n-character set, that set shall be invoked into positions 10/01 to 15/14, and positions 10/00 and 15/15 shall not be used. When a locking-shift function LS1R, LS2R, or LS3R invokes a 96-character or a 96^n-character set, that set shall be invoked into positions 10/00 to 15/15.

8.3.3 Shift status

The shift status is defined as an attribute of a code element G0 to G3 that is in an invoked state, and is identified by the name of the code table area into which the element is invoked i.e. GL or GR. In an 8-bit code only one of G0 to G3 can have a GL shift status, and only one of G1 to G3 can have a GR shift status, at each specific point within a CC-data-element.

Characters from any code element G0 to G3 that does not have a shift status cannot be represented within a CC-data-element (except by the method of 8.4) until that code element has been invoked; i.e. at any given point within a CC-data-element not more than two of the code elements G0 to G3 can be in an invoked state, one in the GL area and one in the GR area.

NOTE 25

The same code element G1, G2, or G3 may be invoked in GL and GR simultaneously, but this is not recommended.

At the beginning of information interchange either locking-shift functions shall be used to establish the initial GL and GR shift status, or the shift status at that point may be undefined (see 10.1).

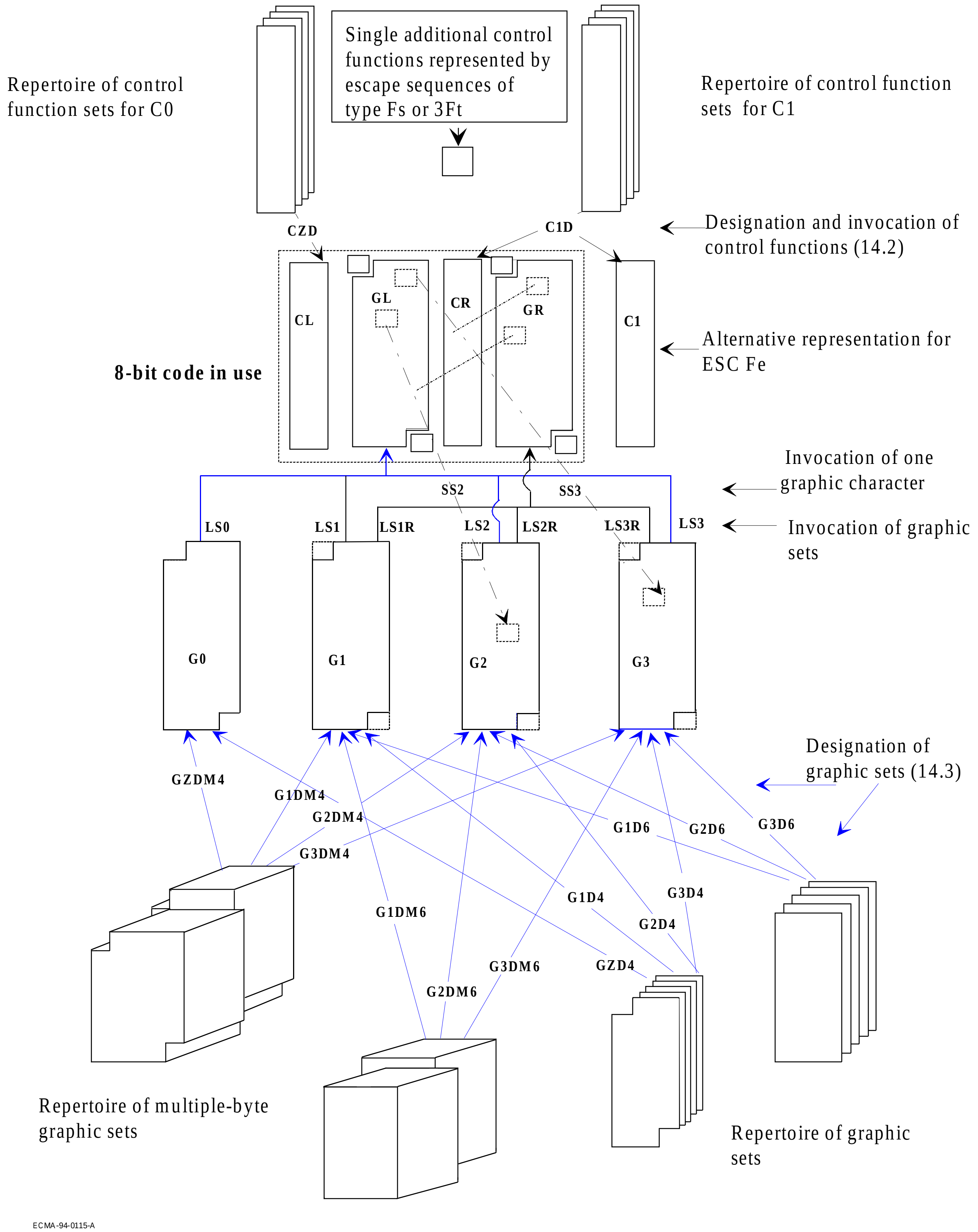


Figure 6 - Structure of 8-bit codes

8.3.4 Interactions of locking-shift functions

When one of the code elements G0 to G3 has GL or GR shift status, the occurrence of the corresponding locking-shift function shall have no effect. The occurrence of a locking-shift function shall have no effect on the meaning of the following bit combinations:

- those representing control characters in columns 00 and 01, or columns 08 and 09;
- those included in any escape sequence;
- those representing the character invoked by an SS2 or SS3 control function (8.4).

An identified graphic character set that has been designated as a code element G0 to G3 may be invoked repeatedly until a different identified set is designated as that code element.

8.4 Invocation of single graphic characters by means of shift functions

SINGLE-SHIFT TWO and SINGLE-SHIFT THREE

If either of the single-shift functions SS2 or SS3 are included in a code (10.1), then either the GL area or the GR area shall be identified as the single-shift area. This identification shall be stated in the specification of the version of the code (10.1).

Each of the single-shift functions SS2 and SS3 shall invoke a single character from the specific character set that has been identified as the respective code element G2 or G3. The graphic character set from which the single character is invoked shall be the set that was last designated as the respective element G2, or G3 of the code.

When such a shift function occurs, the immediately following bit combination (or sequence of n bit combinations for a multiple-byte set) in the CC-data-element shall represent the corresponding character from the set that is designated as the code element G2 or G3 respectively. The shift status that was in effect immediately before the occurrence of the single-shift function shall continue in effect immediately after the bit combination (or sequence) that represents the invoked single graphic character.

When a single-shift function SS2 or SS3 invokes a character from a 94-character or a 94^n-character set the immediately following one or n bit combinations respectively shall be in the range from 02/01 to 07/14 if GL is the single-shift area, and shall be in the range from 10/01 to 15/14 if GR is the single-shift area. When a single-shift function SS2 or SS3 invokes a character from a 96-character or a 96^n-character set the immediately following one or n bit combinations respectively shall be in the range from 02/00 to 07/15 if GL is the single-shift area, and shall be in the range from 10/00 to 15/15 if GR is the single-shift area.

8.5 Invocation of sets of control functions

8.5.1 Invocation of the C0 code element

The designation of a primary set of control functions as the C0 code element of the code shall also invoke that set in the CL area of the code table. When C0 is invoked, each control function (character) in that set shall be represented by the corresponding bit combination from columns 00 and 01 of the code table, i.e. positions 00/00 to 01/15.

If C0 has not been invoked the assumption is made that the control character ESCAPE is available and is represented by bit combination 01/11.

The control characters LS0 and LS1 shall be included in the C0 code element of any specific 8-bit code that makes use of these shift functions (see 8.3).

NOTE 26

A primary set that includes LS0 and LS1 is specified in ECMA-48.

8.5.2 Invocation of the C1 code element

The designation of a supplementary set of control functions as the C1 code element of an 8-bit code shall also invoke that set. When a C1 code element is invoked, either:

- it shall be invoked in the CR area of the code table, or
- it shall be invoked into escape sequences of type ESC Fe (see 13.2.1), and bit combinations in the CR area shall not be used.

Only one of these two alternative forms of invocation shall be used in any version (see 10.1) of an 8-bit code that includes the C1 code element. The form of invocation used shall be identified in the specification of the version. The other form of invocation shall not be used in that code.

The control functions SS2 and SS3 shall be included in the C1 code element of any specific code that makes use of these shift functions (see 8.4).

NOTE 27

A supplementary set that includes SS2 and SS3 is specified in ECMA-48. A supplementary set that includes only SS2 and SS3, and no other control functions, has been registered in the ISO International Register of Coded Character Sets (see annex B).

9 Structure of 7-bit codes

9.1 Code table layout for 7-bit codes

A 7-bit code shall have a structure which is based on a 7-bit code table arranged in separate areas as follows (see figure 7):

- columns 00 and 01 shall be the CL area,
- columns 02 to 07 shall be the GL area.

The bit combinations in these areas shall be used to represent characters as follows:

- CL, a primary set of control characters;
- GL, either the characters SPACE (6.2.3) and DELETE (6.2.1) and a set of 94 or 94^n graphic characters, or a set of 96 or 96^n graphic characters;

NOTE 28

In figure 7 positions 02/00 and 07/15 may be used to represent any graphic characters when a 96-character or 96^n-character set is invoked in the GL area.

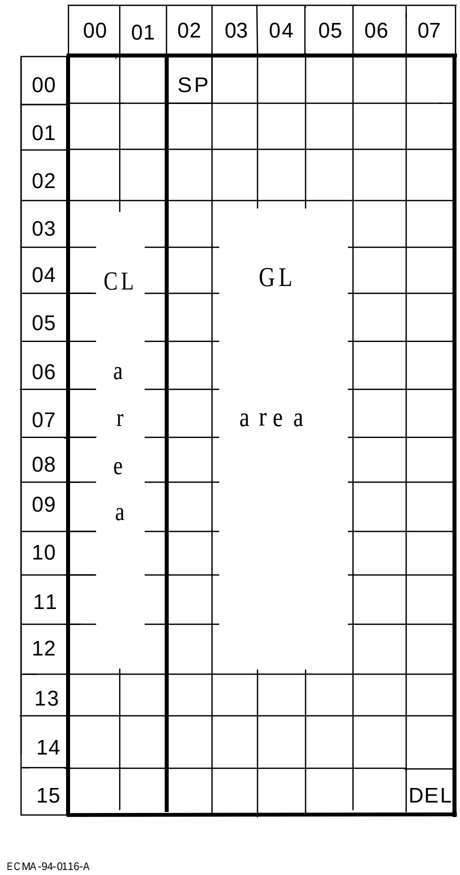


Figure 7 - Areas of 7-bit code table

9.2 Elements and structure of the code

A 7-bit code shall include a C0 and a G0 code element, and may also include some or all of the code elements C1, G1, G2, and G3 (7.2). If more than one of G0 to G3 are included, it shall include appropriate shift functions (7.3) to enable the graphic characters in those sets to be invoked.

A 7-bit code may include one or more single additional control functions (6.5), or code-identification functions (7.4).

If more than four sets of graphic characters, or more than two sets of control functions, are required in a specific 7-bit code, designation functions may be used within that code to change the specific character sets that are designated as code elements G0 to G3, C0, or C1 (14.1).

The structure of the code is illustrated in figure 8.

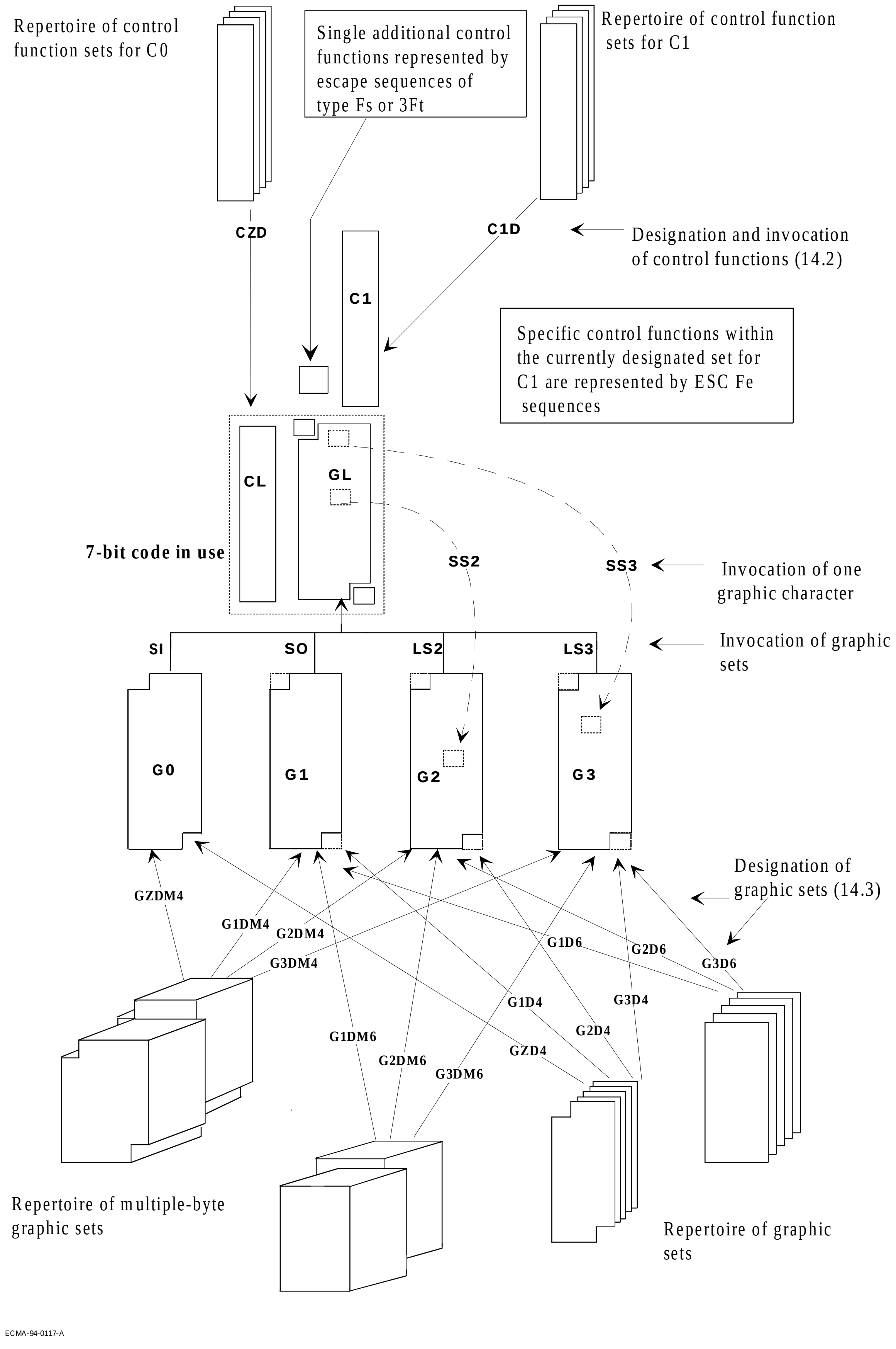


Figure 8 - Structure of 7-bit codes

9.3 Invocation of graphic character sets by means of shift functions

9.3.1 SHIFT-IN, SHIFT-OUT, LOCKING-SHIFT TWO, and LOCKING-SHIFT THREE

Each of the four locking-shift functions SI, SO, LS2, and LS3 (7.3) shall invoke in the GL area the specific graphic character set that has been identified as the respective code element G0, G1, G2, or G3. The graphic character set that is invoked shall be the set that was last designated as the respective element G0, G1, G2, or G3 of the code. When such a set has been invoked the characters in that set shall be represented by the corresponding bit combinations (or sequences of n bit combinations for a multiple-byte set) in the GL area of the code table.

When a locking-shift function SI, SO, LS2, or LS3 invokes a 94-character or 94^n-character set, that set shall be invoked into positions 02/01 to 07/14, SPACE shall be invoked into position 02/00, and DELETE shall be invoked into position 07/15. When a locking-shift function SO, LS2, or LS3 invokes a 96-character or 96^n-character set, that set shall be invoked into positions 02/00 to 07/15. The characters SPACE and DELETE cannot be represented while such a set is in an invoked state.

9.3.2 LOCKING SHIFT ONE RIGHT, TWO RIGHT, and THREE RIGHT

The three shift functions LS1R, LS2R, and LS3R, which are specified in 8.3.2 for use in 8-bit codes, may also be used in a 7-bit code. They shall only be used in a 7-bit code when their meaning must be preserved during transformation of CC-data-elements from an 8-bit to a 7-bit code (see 11.1). When used in a 7-bit code LS1R, LS2R, and LS3R shall have the same effects as SO, LS2, and LS3 respectively.

9.3.3 Shift status

The shift status is defined as an attribute of a code element G0 to G3 that is in an invoked state. In a 7-bit code only one of G0 to G3 can have a shift status at each specific point within a CC-data-element.

Characters from any code element G0 to G3 that does not have a shift status cannot be represented within a CC-data-element (except by the method of 9.4) until that code element has been invoked; i.e. at any given point within a CC-data-element not more than one of the code elements G0 to G3 can be in an invoked state.

At the beginning of information interchange either locking-shift functions shall be used to establish the initial shift status, or the shift status at that point may be undefined (see 10.1).

9.3.4 Interactions of locking-shift functions

When one of the code elements G0 to G3 has shift status, the occurrence of the corresponding locking-shift function shall have no effect. The occurrence of a locking-shift function shall have no effect on the meaning of the following bit combinations:

– those representing control characters in columns 00 and 01;
– those included in any escape sequence;
– those representing the character invoked by an SS2 or SS3 control function (9.4).

An identified graphic character set that has been designated as a code element G0 to G3 may be invoked repeatedly until a different identified set is designated as that code element.

9.4 Invocation of single graphic characters by means of shift functions

SINGLE-SHIFT TWO and SINGLE-SHIFT THREE

Each of the single-shift functions SS2 and SS3 shall invoke a single character from the specific character set that has been identified as the respective code element G2 or G3. The graphic character set from which the single character is invoked shall be the set that was last designated as the respective element G2, or G3 of the code.

When such a shift function occurs, the immediately following bit combination (or sequence of n bit combinations for a multiple-byte set) in the CC-data-element shall represent the corresponding character from the set that is designated as the code element G2 or G3 respectively. The shift status that was in effect immediately before the occurrence of the single-shift function shall continue in effect immediately after the bit combination (or sequence) that represents the invoked single graphic character.

When a single-shift function SS2 or SS3 invokes a character from a 94-character or a 94^n-character set the immediately following one or n bit combinations respectively shall be in the range from 02/01 to 07/14. When a single-shift function SS2 or SS3 invokes a character from a 96-character or a 96^n-character set the immediately following one or n bit combinations respectively shall be in the range from 02/00 to 07/15.

9.5 Invocation of sets of control functions

9.5.1 Invocation of the C0 code element

The designation of a primary set of control functions as the C0 code element of the code shall also invoke that set in the CL area of the code table. When C0 is invoked, each control function (character) in that set shall be represented by the corresponding bit combination from columns 00 and 01 of the code table, i.e. positions 00/00 to 01/15.

If C0 has not been invoked the assumption is made that the control character ESCAPE is available and is represented by bit combination 01/11.

The control characters SI and SO shall be included in the C0 code element of any specific 7-bit code that makes use of these shift functions (see 9.3).

NOTE 29

A primary set that includes SI and SO is specified in ECMA-48.

9.5.2 Invocation of the C1 code element

The designation of a supplementary set of control functions as the C1 code element of a 7-bit code shall also invoke that set into escape sequences of type ESC Fe (see 13.2.1).

The control functions SS2 and SS3 shall be included in either the C1 or the C0 code element of any specific code that makes use of these shift functions (see 9.4).

NOTE 30

A supplementary set that includes SS2 and SS3 is specified in ECMA-48. A supplementary set that includes only SS2 and SS3, and no other control functions, has been registered in the ISO International Register of Coded Character Sets (see annex B).

10 Versions and levels of implementation

10.1 Versions

A version of the code structure specified in this Standard is an identified selection of the facilities specified in clause 8 or in clause 9.

A version will generally be the subject of a specification document which states the selection of facilities comprised within that version. Such a document provides a basis for agreement between the interchanging parties.

A version may optionally include identification of the initial conditions applicable at the beginning of information interchange, in particular:

– the identification of all graphic character sets and control function sets that are to be used,
– the initial designation status of such sets, and
– the shift status.

Thus a version may be, but need not be, a fully specified code.

If a version is not a fully specified code the means by which identifications are communicated between the interchange parties prior to information interchange shall be defined in the specification of the version.

NOTE 31

A version of this Standard differs in principle from a version of a standard such as ECMA-43 since that standard requires a version to be a fully specified code.

10.2 Identification of code structure facilities and character sets

CC-data-elements conforming to a version of this Standard are intended to form all or part of a composite unit of coded information that is interchanged between a sender and a recipient. The identification of the version of this Standard that has been adopted by the originator, and any other identifications needed to define a fully specified code, shall also be available to the recipient. The route by which such identification is communicated to the recipient is outside the scope of this Standard.

However some standards for interchange of coded information may permit, or require, that the coded representation of the identifications applicable to the CC-data-elements forms part of the interchanged information. Such coded representations may form all or part of a distinct identification data element, or may be embedded in a CC-data-element, in accordance with the relevant standard. The code-identification functions for announcement (15.2) and designation (14) specified in this Standard may be used for this purpose. Object identifiers in accordance with ISO 8824 (ASN.1) may also be used for this purpose - see annex A.

10.3 Levels of implementation

The following selections of code structure facilities are defined here because of their wide applicability. Their adoption is intended to improve the extent of compatibility achievable in information interchange.

10.3.1 8-bit codes

Level 1 - Elementary 8-bit Code

A Level 1 version of the 8-bit code structure of this Standard shall comprise:

- a C0 code element,
- a G0 code element having GL shift status,
- the characters SPACE and DELETE,
- optionally a C1 code element in the CR area,
- a G1 code element having GR shift status.

The designation functions for C0, C1, G0, and G1 may optionally be used at the beginning of information interchange only. The use of the locking-shift functions LS0 and LS1R is not required.

NOTE 32

ECMA-43 (Level 1) is in accordance with Level 1.

Level 2 - Multiple Pre-Designated Character Sets, without Locking Shifts

A Level 2 version of the 8-bit code structure of this Standard shall comprise:

- the facilities of Level 1,
- either or both of the G2 and G3 code elements,
- one or both of the shift functions SS2 and SS3, with GL identified as the single-shift area.

The designation functions for G2 and G3 may optionally be used at the beginning of information interchange only.

NOTE 33

ECMA-43 (Level 2) is in accordance with Level 2.

Level 3 - Multiple Pre-Designated Character Sets, with Locking Shifts

A Level 3 version of the 8-bit code structure of this Standard shall comprise:

- the facilities of Level 2,
- one or more of the shift functions LS0, LS1, LS2, LS3, LS1R, LS2R, LS3R.

NOTE 34

ECMA-43 (Level 3) is in accordance with Level 3.

Level 4 - Redesignation of Graphic Character Sets Within a Code

A Level 4 version of the 8-bit code structure of this Standard shall comprise:

- the facilities of Level 3,
- the use of the graphic set designation functions (14.3) at any point during information interchange.

The announcement function (15.2) may optionally be used at the beginning of information interchange only.

NOTE 35

In a Level 4 version, the coded representation of one or more characters may change when one of the code elements G0, G1 or G2 is redesignated, if the requirement for unique coding applies (see 7.5).

10.3.2 Qualification of levels for 8-bit codes

Levels 2A, 3A, and 4A - GR as single-shift area

In a code that is according to Levels 2, 3, or 4, except that GR is identified as the single-shift area, then the corresponding levels are identified as Level 2A, 3A, and 4A.

Levels 1C, 2C, 3C and 4C - C1 invoked as ESC Fe

In a code that is according to Levels 1, 2, 3, or 4, except that the functions in the C1 code element are represented by ESC Fe sequences, instead of by bit combinations in columns 08 and 09, the corresponding implementation levels are identified as Level 1C, 2C, 3C, and 4C.

If both of the above qualifications apply, the levels are identified as Level 2AC, 3AC, and 4AC.

10.3.3 7-bit codes

Level 1 - Elementary 7-bit Code

A Level 1 version of the 7-bit code structure of this Standard shall comprise:

- a C0 code element,
- a G0 code element having shift status,
- the characters SPACE and DELETE.

The designation functions for C0 and G0 may optionally be used at the beginning of information interchange only. The use of the locking-shift function SI is not required.

NOTE 36

ECMA-6 is in accordance with Level 1.

Level 2 - Multiple Pre-Designated Character Sets With Two Locking Shifts

A Level 2 version of the 7-bit code structure of this Standard shall comprise:

- the facilities of Level 1,
- optionally a C1 code element,
- the G1 code element,
- the shift functions SO and SI,
- one or both of the G2, and G3 code elements,
- one or both of the shift functions SS2 and SS3.

The designation functions for C1, G1, G2, and G3 may optionally be used at the beginning of information interchange only.

Level 3 - Multiple Pre-Designated Character Sets With Four Locking Shifts

A Level 3 version of the 7-bit code structure of this Standard shall comprise:

- the facilities of Level 2,
- one or both of the locking shift functions LS2 and LS3.

Level 4 - Redesignation of Graphic Character Sets Within a Code

A Level 4 version of the 7-bit code structure of this Standard shall comprise:

- the facilities of Level 3,
- the use of the graphic set designation functions (14.3) at any point during information interchange.

The announcement function (15.2) may optionally be used at the beginning of information interchange only.

NOTE 37

In a Level 4 version, the coded representation of one or more characters may change when one of the code elements G0, G1 or G2 is redesignated, if the requirement for unique coding applies (see 7.5).

NOTE 38

The previous edition of this Standard defined three "levels of compatibility" with ECMA-6 which could apply to a code with code extension facilities. They have been replaced here by the above set of four levels which have a wider applicability.

11 Transformation between 8-bit and 7-bit codes

11.1 Transformation from 8-bit to 7-bit codes

Information conforming to any 8-bit code that is in accordance with this Standard may be transformed to an equivalent 7-bit coded form when required. The transformation rules depend on which facilities of code structure are included in the 8-bit code. Identification of those facilities may be achieved by the use of the announcement function defined in 15.2.

Within a CC-data-element any 8-bit combination in which b_8 is ZERO has an equivalent 7-bit form comprising b_7 to b_1 of the 8-bit form. Where the bit combination has a b_8 of ONE (i.e. it is from columns 08 to 15 of the code table) further transformation is required. This applies to the following coded representations:

- control functions from C1 if represented in columns 08 and 09,
- graphic characters from whichever of G1, G2, or G3 has GR shift status,
- single graphic characters invoked by SS2 or SS3 when they are represented by bit combinations from columns 10 to 15 (i.e. GR has been identified as the single-shift area).

In the equivalent 7-bit form the corresponding coded representations are as follows:

- a control function from the C1 set is represented by an ESC Fe sequence,
- a graphic character from the set that has GR shift status is represented by b_7 to b_1 of the corresponding 8-bit combination (or sequence of n 8-bit combinations for a multiple-byte set), and appropriate extra shift functions are present,
- a single graphic character invoked by SS2 or SS3 is represented by b_7 to b_1 of the corresponding bit combinations from columns 02 to 07.

11.2 Transformation from 7-bit to 8-bit codes

Information conforming to any 7-bit code that is in accordance with this Standard may be transformed to an equivalent 8-bit coded form when required. The transformation rules depend on which facilities of code structure are included in the 7-bit code. Identification of those facilities may be achieved by the use of the announcement function defined in 15.2.

Within a CC-data-element any 7-bit combination has an equivalent 8-bit form comprising b_7 to b_1 of the 7-bit form, together with a b_8 of ZERO.

Further transformations may be made to obtain a fuller exploitation of the capability of an 8-bit code, in particular to make use of the following coded representations:

- control functions from the C1 set represented in columns 08 and 09,
- graphic characters from a G1, G2, or G3 set having GR shift status, and the elimination of some shift functions that are consequently redundant.
- single graphic characters invoked by SS2 or SS3 when GR has been identified as the single-shift area for the code.

This Standard does not specify the rules for such transformations. The announcement function ACS 5 may be used to indicate that all shift functions have been preserved.

If a 7-bit CC-data-element originated from a previous transformation of an equivalent 8-bit form, the locking-shift functions LS1R, LS2R, and LS3R may be present. Further transformation of the coded representations of characters from G1, G2, and G3 sets may be required in consequence of the differences of specification of these functions in 7-bit and 8-bit codes. The announcement function ACS 10 and ACS 11 may be used to indicate which specification is intended to apply.

Section 3 - Code identification and escape sequences

12 Code-identification functions

12.1 Purposes of code-identification functions

Code-identification functions to provide the following facilities are specified in this Standard:

– designation of sets of graphic characters,
– designation of sets of control functions or characters,
– identification of the revision number of registered character sets,
– announcement of the byte-size (8-bit or 7-bit) of the code and the facilities of code structure used,
– designation of, and return from, other coding systems that are not in accordance with this Standard,

An associated control function is also specified:

– data delimiter to indicate a return to an outer level of data structure.

These functions are specified in clauses 14 and 15 below.

12.2 Relationship to escape sequences

Escape sequences provide the coded representations of code-identification functions. Their structure is specified in the following clause.

NOTE 39

Escape sequences also provide the coded representations of supplementary sets of control functions (6.4.3) and of single additional control functions (6.5).

13 Structure and use of escape sequences

13.1 Structure of escape sequences

An escape sequence shall consist of two or more bytes. In an 8-bit code a byte shall be an 8-bit combination. In a 7-bit code a byte shall be a 7-bit combination.

The first byte of an escape sequence shall be the bit combination representing the ESCAPE character and the last shall be known as the Final Byte. An escape sequence may also contain one or more bytes known as Intermediate bytes.

The function represented by an escape sequence shall be determined by its Intermediate byte(s), if any, and by its Final Byte.

Intermediate bytes shall be any of the 16 positions of column 02 of the code table; they are denoted by the symbol I.

Final bytes shall be any of the 79 positions of columns 03 to 07 of the code table excluding position 07/15; they are denoted by the symbol F.

NOTE 40

Although, in this Standard, escape sequences are specified in terms of bytes or positions in the code table, the meaning of an escape sequence is determined only by its bit combinations and it is unaffected by any meaning assigned to those bit combinations taken individually.

Bit combinations in columns 00 and 01 and the bit combination 07/15 shall not be used as either Intermediate or Final bytes to construct an escape sequence. In an 8-bit code bit combinations in columns 08 to 15 also shall not be so used.

NOTE 41

As these prohibited bytes may appear in an escape sequence in error, it may be necessary within an application to provide methods of identifying such a situation and of recovering from it, but this is not covered by this Standard.

13.2 Types of escape sequences

13.2.1 Indication of type

The first, or only, byte that follows the ESCAPE character in an escape sequence shall indicate the type of the escape sequence. The number of the code table column from which the bit combination of this byte is taken shall identify the type in accordance with the table 3.a. Each type of escape sequence shall be used for the purpose shown in table 3.a or in table 3.b.

Table 3.a - Types of escape sequence and corresponding indicators

Column No.	Type	Function represented
00 and 01	-	(shall not be used)
02	nF	(see table 3.b)
03	Fp	Private control function (see 6.5.3)
04 and 05	Fe	Control function in the C1 set (see 6.4.3)
06 and 07 except 07/15	Fs	Standardized single control function (see 6.5.1)

In escape sequences of types Fp, Fe, and Fs the second byte (the type-indicator) shall also be the Final Byte. The notations Fp, Fe, and Fs respectively are also used when referring to the coded representations of control functions.

The type "nF" in the above table indicates escape sequences of the series of types whose names are of the form nF where n may take any value from 0 to 15, as listed in table 3.b. Such escape sequences shall be of the form ESC I .. F where the notation ".." indicates that more than one Intermediate Byte may appear in the sequence.

13.2.2 Escape Sequences of types nF

Escape sequences of the types nF shall be used to represent various code-identification functions, except for type 3F which shall represent control functions. Table 3.b shows, for each type, the type name, the first I byte (the type-indicator byte), and the corresponding function represented by it. It also shows whether the sequence may include further I bytes, and their purpose if present. (The key to the notation for the entries in the column headed "2nd I byte" follows the table.)

Within each of the above types the escape sequences are of two subtypes, depending on the range of the F byte. The table below shows for each subtype the range of F and the assigned use.

Range of F	Notation	Use of F
columns 00 to 02	-	(shall not be used)
column 03	Fp	Reserved for Private use (16 positions, see 13.3.3)
columns 04 to 07 except 07/15	Ft	Standardised purposes (63 positions, see 13.3.1 and 13.3.2)

Table 3.b - Escape sequences of type nF and corresponding functions

Type	First I Byte	2nd I Byte*		Function represented
		Rule	Use	
0F	02/00	N	-	ANNOUNCE CODE STRUCTURE(15.2)
1F	02/01	O	R	C0-DESIGNATE (14.2)
2F	02/02	O	R	C1-DESIGNATE (14.2)
3F	02/03	O	R	Single control functions (6.5.2 and 6.5.3)
4F	02/04	Y	S	Designation of multiple-byte graphic character sets (see 13.2.3 and 14.3)
5F	02/05	O	R,S	DESIGNATE OTHER CODING SYSTEM (15.4)
6F	02/06	N	-	IDENTIFY REVISED REGISTRATION (14.5)
7F	02/07	N	-	(reserved for future standardisation)
8F	02/08	O	R,S	G0-DESIGNATE 94-SET (14.3)
9F	02/09	O	R,S	G1-DESIGNATE 94-SET (14.3)
10F	02/10	O	R,S	G2-DESIGNATE 94-SET (14.3)
11F	02/11	O	R,S	G3-DESIGNATE 94-SET (14.3)
12F	02/12	N	-	(reserved for future standardisation)
13F	02/13	O	R,S	G1-DESIGNATE 96-SET (14.3)
14F	02/14	O	R,S	G2-DESIGNATE 96-SET (14.3)
15F	02/15	O	R,S	G3-DESIGNATE 96-SET (14.3)

*** Key to “2nd I byte” entries in table 3.b**

When the Final Byte of an escape sequence is Fp the 2nd (and further) I Bytes are not specified by this Standard. When the Final Byte is Ft the entries in the "Rule" and "Use" columns of table 3.b indicate the following provisions relating to the 2nd (and any further) I bytes. Any bit combinations not specified explicitly below shall be reserved for future standardisation.

Rule:

N 2nd I byte shall not be present (reserved for future standardisation);
O 2nd (and further) I bytes may be present for use as indicated under "Use";
Y 2nd I byte shall be present for use as indicated under "Use".

Use:

R Registration procedures may make use of bit combinations 02/01 to 02/03 to supplement the number of bit combinations available for F (see 14.1). Further I bytes in the range 02/00 to 02/15 may also be so used.

S Standardised meanings are assigned to one or more bit combinations, as shown below.

Escape Sequence Type	Bit combination
4F	(for 2nd I byte see 13.2.3)
4F	02/00 of 3rd I byte (14.4)
5F	02/15 of 2nd I byte (15.4)
8F to 11F and 13F to 15F	02/00 of 2nd I byte (14.4)

13.2.3 Escape Sequences of type 4F

Table 4 below shows the assignment of the 2nd I byte in escape sequences of type 4F, and the corresponding function represented by the sequence.

Table 4 - Escape sequences of type 4F and corresponding functions

2nd I Byte	Function represented (see 14.3)
02/08	G0-DESIGNATE MULTIBYTE 94-SET
02/09	G1-DESIGNATE MULTIBYTE 94-SET
02/10	G2-DESIGNATE MULTIBYTE 94-SET
02/11	G3-DESIGNATE MULTIBYTE 94-SET
02/13	G1-DESIGNATE MULTIBYTE 96-SET
02/14	G2-DESIGNATE MULTIBYTE 96-SET
02/15	G3-DESIGNATE MULTIBYTE 96-SET

NOTE 42

The 2nd I byte 02/08 is omitted when F is 04/00, 04/01 or 04/02 (see 14.3.2, table 6, Note).

In escape sequences of type 4F a third I byte may be present, with assignments of bit combinations as for "Use" R,S (see key to table 3 above).

13.2.4 Summary

Table 5 summarises the assignments of the Intermediate bytes in the escape sequences. The shaded area denotes the combinations reserved for future standardization.

13.2.5 Notation of escape sequences

In this Standard the notation used for escape sequences is as follows, where x/y is as defined in 5.1, and F indicates Fp or Ft as defined in 13.2.2:

Type	**Notation**
Fe, Fp, and Fs	ESC x/y
nF	ESC x/y F

Table 5 - Intermediate bytes of escape sequences

SECOND INTERMEDIATE BYTE
When the Final Byte is Registered or Standardized (F = Ft)

Code Position	FIRST INTERMEDIATE BYTE: Purpose of Function		02/00	02/01 02/02 02/03	02/04 02/05 02/06 02/07	02/08	02/09	02/10	02/11	02/12	02/13	02/14	02/15
02/00	Announcers												
02/01	Designate sets of control functions	C0 set		Registration Authority Use supplementary to the Final Byte									
02/02	Designate sets of control functions	C1 set		Registration Authority Use supplementary to the Final Byte									
02/03	Single control functions												
02/04	Designate sets of multiple-byte graphic characters					94^n set to G0 set	94^n set to G1 set	94^n set to G2 set	94^n set to G3 set		96^n set to G1 set	96^n set to G2 set	96^n set to G3 set
02/05	Designate other coding systems			Registration Authority Use									No Standard Return
02/06	Identify revision of registration												
02/07													
02/08	Designate sets of 94 graphic characters	G0 set	D R C S	Registration Authority Use supplementary to the Final Byte									
02/09	Designate sets of 94 graphic characters	G1 set	D R C S	Registration Authority Use supplementary to the Final Byte									
02/10	Designate sets of 94 graphic characters	G2 set	D R C S	Registration Authority Use supplementary to the Final Byte									
02/11	Designate sets of 94 graphic characters	G3 set	D R C S	Registration Authority Use supplementary to the Final Byte									
02/12													
02/13	Designate sets of 96 graphic characters	G1 set	D R C S	Registration Authority Use supplementary to the Final Byte									
02/14	Designate sets of 96 graphic characters	G2 set	D R C S	Registration Authority Use supplementary to the Final Byte									
02/15	Designate sets of 96 graphic characters	G3 set	D R C S	Registration Authority Use supplementary to the Final Byte									

ECMA-94-0118-A

13.3 Specific meanings of escape sequences

13.3.1 Registration of Final Bytes

In an escape sequence of type Fs or of type nF, except for type 0F and except when the SELECT-DRCS indicator is present (see 14.4), the Final Byte Ft shall be reserved for registration.

Where the Final Byte of a type of escape sequence is reserved for registration its meaning, and that of any I bytes to supplement it, will be specified using the procedures established by ISO 2375. That Standard shall be followed in preparing and maintaining a register of escape sequences and their meanings. Allocation of such Final and, where necessary, Intermediate bytes is carried out by the Registration Authority in accordance with ISO 2375 (see annex B), except as stated in 13.3.2.

13.3.2 Final Bytes specified in this Standard

Standardised meanings are assigned to one or more bit combinations for the Final Byte F, as shown below.

Escape sequence type	Bit combination
0F	(see 15.2)
1F, 2F, 4F	07/14 (14.1)
5F	04/00 (15.4)
8F to 11F and 13F to 15F	07/14 (14.1)

13.3.3 Private use

In any escape sequence a Final Byte Fp (i.e. from column 03) shall be reserved for private use. Escape sequences for private use are not subject to registration under ISO 2375. They are defined by agreement between the interchange parties.

Where the SELECT-DRCS indicator is present in an escape sequence of type nF the Final Byte Ft shall also be for private use (14.4).

NOTE 43

The implementors of any private escape sequence described as such in this Standard are alerted to the fact that other implementors may give different meanings to the same escape sequence or may use different escape sequences to mean the same thing. Furthermore, such meanings may subsequently be assigned to registered escape sequences. Interchange parties are warned that the use of such private escape sequences may reduce their capability to interchange data subsequently.

14 Designation of sets of graphic characters and control functions

14.1 Designation functions

Designation functions are provided to enable identified sets of coded graphic characters, and identified sets of coded control functions, to be designated as elements of a specific code. Any such set that is designated shall be either:

- registered in the ISO International Register of Coded Character Sets (see annex B), or
- a set for private use, or
- a dynamically redefinable (graphic) character set (DRCS, see 14.4).

A function to identify a revision of a registered character set is also provided.

A registered set, or a set for private use, shall be of one of the following types:

- a primary set of control functions for C0 (see 6.4.2)
- a supplementary set of control functions for C1 (see 6.4.3)
- a 94-character set of graphic characters (see 6.3.1)
- a 96-character set of graphic characters (see 6.3.1)
- a 94^n-character set of graphic characters (see 6.3.1)
- a 96^n-character set of graphic characters (see 6.3.1)

Separate designation functions are provided for designating a primary set for C0 and a supplementary set for C1. For graphic character sets separate designation functions are provided according to the type (i.e. size) of set to be designated and the code element G0 to G3 into which the set is to be designated.

Designation functions shall be represented by escape sequences as defined in clause 13. For each registered set a Final Byte F (=Ft), and where necessary one or more associated Intermediate byte(s) I to supplement the F byte, is allocated in the register to identify the set. This (sequence of) byte(s) shall be the F byte (and associated I bytes, if any) in the escape sequence representing the function that designates that set.

NOTE 44

Such allocations of F (and associated I, if any) are taken from a separate series of Final bytes for each of the six types of sets indicated above. In consequence a designation function can only designate a set of suitable type as a particular code element.

The Final Byte 07/14 shall be reserved to identify the empty set of each type, i.e. the set that does not contain any character. If an empty set is designated the bit combinations corresponding to the code table positions of the set shall not be used, i.e. the (non-existent) characters shall not be invoked.

A Final Byte F (= Fp) shall identify a set for private use (see 13.3.3).

14.2 Designation of sets of control functions (CZD, C1D)

14.2.1 Purpose

Two functions are specified here for designating sets of control functions as elements of a code. They designate respectively the C0 and C1 code elements. They may be used for the following purposes:

a) at the beginning of information interchange, to identify the specific control function sets to be used as C0 and C1 code elements;

b) if more than two sets of control functions are required for use in a specific code;

c) if a succession of different control function sets is required to be designated as a particular code element C0 or C1.

In cases b) and c) the appropriate designation functions are regarded as part of the code structure.

NOTE 45

ECMA-48 specifies an alternative method for providing additional control functions within a code. That method is preferred when there is a need to avoid superseding one set of control functions by another.

The function IDENTIFY-REVISED-REGISTRATION (IRR), specified in 14.5, may be used in conjunction with these designation functions.

14.2.2 Designation of C0

Name: C0-DESIGNATE Acronym: CZD Coded representation: ESC 02/01 F

CZD shall designate an identified primary set of control functions as the C0 element of a specific code. The set shall be identified by the Final Byte F. The designation of a set as a C0 code element shall immediately invoke that set and shall supersede the previously designated set.

14.2.3 Designation of C1

Name: C1-DESIGNATE Acronym: C1D Coded representation: ESC 02/02 F

C1D shall designate an identified supplementary set of control functions as the C1 element of a specific code. The set shall be identified by the Final Byte F. The designation of a set as a C1 code element shall immediately invoke that set and shall supersede the previously designated set.

14.3 Designation of sets of graphic characters (GnDm and GnDMm)

14.3.1 Purpose

Fourteen functions are specified here for designating sets of graphic characters. The specifications of these functions are similar to each other, differing only as regards the type of set designated and the code element to which the designation applies. They are listed in table 6.

These functions may be used for the following purposes:

a) at the beginning of information interchange, to identify the specific graphic character sets to be used as G0, G1, G2, and G3 code elements;

b) when more than four sets of graphic characters are required for use in a specific code;

c) if a succession of different graphic character sets is required to be designated as a particular code element G0, G1, G2, or G3.

In cases b) and c) the appropriate designation functions are regarded as part of the code structure.

The SELECT-DRCS indicator, specified in 14.4, may be used with these designation functions. The function IDENTIFY-REVISED-REGISTRATION (IRR), specified in 14.5, may be used in conjunction with these functions.

14.3.2 Specifications

Each row of table 6 summarises the features of one of the functions specified here for designating a graphic character set as an element of a specific code. The entries in a row indicate the following features of the function:

- an acronym for the function, of the form GnDm or GnDMm (n=0 to 3, m=4 or 6),
- the name of the function,
- the type (i.e. size) of graphic character set designated by the function,
- the code element to which the function designates the set,
- the I byte (or the 1st and 2nd I bytes) of the escape sequence representing the function.

Each named function shall designate an identified graphic character set of the indicated type as the indicated code element. Its coded representation shall be an escape sequence of type nF where the I byte(s) are as indicated. The set shall not be a DRCS. It shall be identified by the F byte (and any associated I bytes), see 14.1.

A 94-character graphic set or a 94^n-character graphic set may be designated as a G0, G1, G2, or G3 code element of a specific code. Any type of graphic character set may be designated as a G1, G2, or G3 code element. When so designated the identified set shall supersede the set that was previously designated as that particular code element.

A graphic character set may be designated as one of G0 to G3 regardless of the shift status of that code element. When an identified set is designated as a particular code element then:

- if that code element already has a shift status, the set shall be immediately invoked;
- if that code element does not have a shift status, then the set shall be invoked at the next occasion when the corresponding locking-shift function is used (8.3 and 9.3).

Table 6 - Designation functions for graphic character sets

Acronym	Name	Type of set	Code element	I byte(s)
GZD4	G0-DESIGNATE 94-SET	94	G0	02/08
G1D4	G1-DESIGNATE 94-SET	94	G1	02/09
G2D4	G2-DESIGNATE 94-SET	94	G2	02/10
G3D4	G3-DESIGNATE 94-SET	94	G3	02/11
G1D6	G1-DESIGNATE 96-SET	96	G1	02/13
G2D6	G2-DESIGNATE 96-SET	96	G2	02/14
G3D6	G3-DESIGNATE 96-SET	96	G3	02/15
GZDM4	G0-DESIGNATE MULTIBYTE 94-SET	94^n	G0	02/04 02/08 *
G1DM4	G1-DESIGNATE MULTIBYTE 94-SET	94^n	G1	02/04 02/09
G2DM4	G2-DESIGNATE MULTIBYTE 94-SET	94^n	G2	02/04 02/10
G3DM4	G2-DESIGNATE MULTIBYTE 94-SET	94^n	G3	02/04 02/11
G1DM6	G1-DESIGNATE MULTIBYTE 96-SET	96^n	G1	02/04 02/13
G2DM6	G2-DESIGNATE MULTIBYTE 96-SET	96^n	G2	02/04 02/14
G3DM6	G3-DESIGNATE MULTIBYTE 96-SET	96^n	G3	02/04 02/15

* For the function GZDM4 there is an exception to the above specification. This function shall be represented by ESC 02/04 F with no 2nd I byte when F is 04/00, 04/01, or 04/02.

NOTE 46

The reason for this exception is that the sets registered with Final bytes 04/00, 04/01, and 04/02 were registered at a time when the current version of this Standard (1st Edition) allowed multiple-byte sets to be designated only as G0 sets and specified ESC 02/04 F to represent that designation.

14.3.3 Size indication for multiple-byte sets

For the functions that designate 94^n-character and 96^n-character graphic character sets (i.e. the functions GnDMm), the range of the Final Byte F in the escape sequence that represents the function shall indicate the number of bytes in the coded representation of the characters in the set identified by F, as follows:

Range of F	**Number of Bytes**
columns 00 to 02	(shall not be used)
column 03	2 or more (Note: the set is for private use)
columns 04 and 05	2
column 06	3
column 07	4 or more

14.4 Dynamically redefinable character sets (DRCS)

14.4.1 Purpose

A Dynamically Redefinable Character Set (DRCS) is a set of graphic characters the visual appearance of which is specified and transmitted prior to the time of use. Such specification may be made explicitly or by a reference. These characters may be alphabetic, syllabic, ideographic, special symbols or picture element symbols. Once specified, a DRCS is regarded as a member of the repertoire of graphic character sets that can be designated by appropriate escape sequences as a G0, G1, G2 or G3 code element.

The designation functions specified in 14.3 are used with an added indicator, SELECT-DRCS, to designate a DRCS.

14.4.2 Specification

When the SELECT-DRCS indicator is included in a designation function of 14.3, the function shall designate an identified DRCS of the indicated type as the indicated code element. The coded representation of the function shall be the appropriate escape sequence from table 6. The SELECT-DRCS indicator shall be represented by the bit combination 02/00 in the 2nd I byte for functions GnDm, and in the 3rd I byte for functions GnDMm.

The DRCS shall be identified by the F byte (and any associated I bytes). Each F byte, and the associated I byte(s) if any, shall be allocated by the user.

NOTE 47

It is recommended that the user should allocate the F bytes sequentially, starting with 04/00. Up to 63 94-character or 94^n-character sets, and up to 63 96-character or 96^n-character sets may be identified within the available range of F bytes, without the use of any associated I bytes.

NOTE 48

The indicator SELECT-DRCS is needed to distinguish between a designated set that is a DRCS and a set that is registered, since it implies exact description of the shape or font of the characters in the set.

14.5 Identification of revisions of registered character sets (IRR)

14.5.1 Purpose

A function is provided to identify the revision number of a character set that is registered in the ISO International Register of Coded Character Sets (see annex B).

A revision of a set in the register is only permitted to add a character or characters to a set, and shall be submitted to the Registration Authority as required by ISO 2375, pointing out that the submission is a revision of a registered set. If the proposed revision is not upwards compatible with the existing version it is not accepted, and it shall be registered as an entirely new set with a new Final Byte allocation.

14.5.2 Specification

Name: IDENTIFY REVISED REGISTRATION Acronym: IRR Coded representation: ESC 02/06 F

IRR when used shall immediately precede one of the designation functions specified in 14.2 and 14.3. IRR shall identify the revision number of the character set that is designated by the immediately following designation function.

In the coded representation of IRR the F byte shall identify the revision by its number in the range 1 to 63, where successive numbers starting from 1 shall be represented by successive bit combinations in columns 04 to 07 starting from 04/00.

NOTE 49

The combined use of IRR and a designation function from 14.3 facilitates the recognition, by older devices or systems, of newer versions of character sets.

15 Code announcement and switching

15.1 Summary of functions provided

Various code-identification and control functions are needed by some applications, in addition to the designation functions specified in clause 14. The functions provided here are:

- an announcer to identify the byte-size (8-bit or 7-bit) of the code and the code structure facilities used (15.2);
- a designation to switch to, and return from, other coding systems that are not in accordance with this Standard (15.4);
- a data delimiter to indicate the end of a unit of data that is in accordance with this Standard (15.3);

These functions shall be represented by escape sequences in accordance with clause 13.

15.2 Announcement of code structure facilities (ACS)

15.2.1 Purpose

A code-identification function is provided to announce the code structure facilities from this Standard that are used in the data which follows. It is intended to be embedded within the character coded information at the beginning of an information interchange. The announcer function may be used one or more times in sequence to identify the various facilities used.

15.2.2 Specification

Name: ANNOUNCE CODE STRUCTURE Acronym: ACS Coded representation: ESC 02/00 F

ACS shall identify a specific code structure facility, or a related group of facilities, which are to be used in following data. The facilities that may be identified by ACS are specified here.

NOTE 50

Identification of additional facilities that might be required in future will need a revision of this Standard. There is no provision for registration of such facilities.

In the coded representation of ACS the Final Byte F shall identify one such facility by its number in the range 1 to 62, where successive numbers starting from 1 shall be represented by successive bit combinations in columns 04 to 07 starting from 04/01.

A list of the facilities that may be identified by ACS is given in table 7. Each entry in the list gives a description of the facility (or group), and a pictorial representation where appropriate, together with the corresponding facility number and Final Byte.

Restrictions on ACS

When ACS is used twice or more in a sequence to identify the code structure facilities applicable to a following unit of data, the following facilities, identified by facility number, shall not be used in the combinations shown:

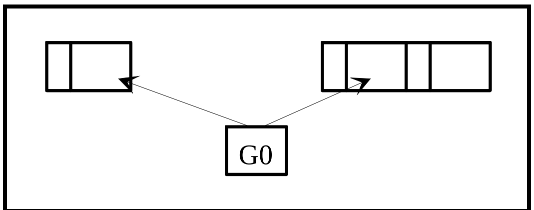
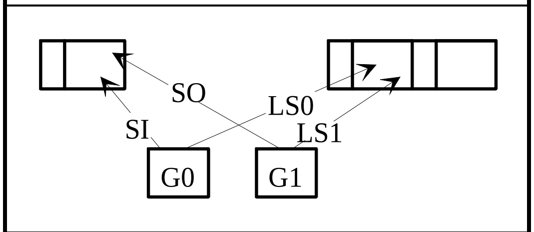
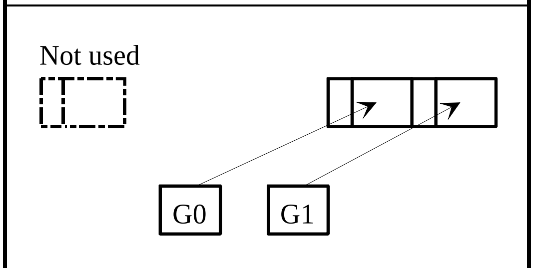
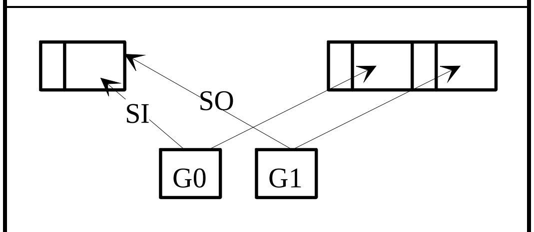
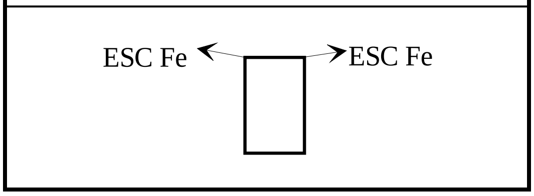
- 1,3, and 4 shall not be used in combination with 16, and 18 to 23;
- 12, 13, and 14 shall not be used together with any other facility.

NOTE 51

In a 7-bit code, facilities number 2 and number 4 provide an identical code structure. Both facilities are provided for those interchange situations in which it is required to differentiate in the 7-bit code between data originating from two types of 8-bit code, viz. those having the G1 code element in columns 02-07 and those having G1 in columns 10-15.

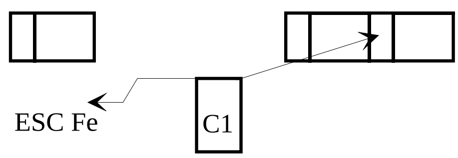
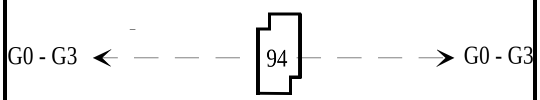
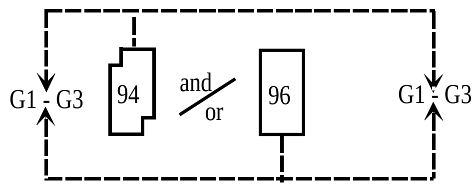
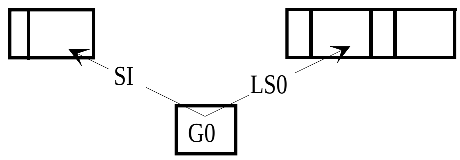
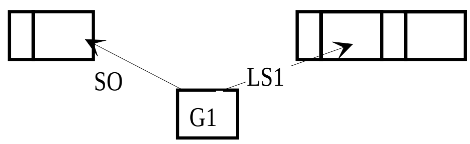
Table 7

Code Structure Facilities for the Announcer Function (ACS)

Facility Number	Final Byte	Facilities utilised	7-bit code	8-bit code
1	04/01	The G0 code element shall be used. Designation of G0 also invokes it into the GL area. No locking-shift functions shall be used. In 8-bit code: the GR area is not used. (*)		
2	04/02	The G0 and G1 code elements shall be used. In 7-bit code: SI invokes G0, and SO invokes G1. In 8-bit code: LS0 invokes G0 to GL, and LS1 invokes G1 to GL. The GR area is not used.		
3	04/03	The G0 and G1 code elements shall be used in an 8-bit code only. Designation of G0 and G1 invoke them to GL and GR respectively. No locking-shift functions shall be used. (*)		
4	04/04	The G0 and G1 code elements shall be used. In 7-bit code: SI invokes G0 and SO invokes G1. In 8-bit code: designation of G0 and G1 also invoke them into GL and GR respectively; no locking-shift functions shall be used. (*)		
5	04/05	Full preservation of shift functions is maintained when transforming data between 7-bit and 8-bit codes.	See 11	
6	04/06	The C1 code element shall be used. In 7-bit and 8-bit codes each C1 control function shall be represented by the ESC Fe sequence.		

(*) see Restrictions on ACS (in 15.2.2)

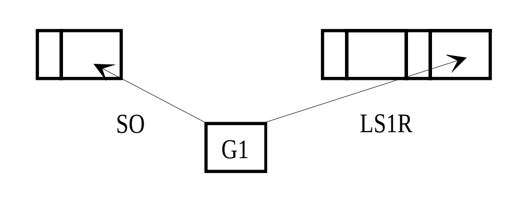
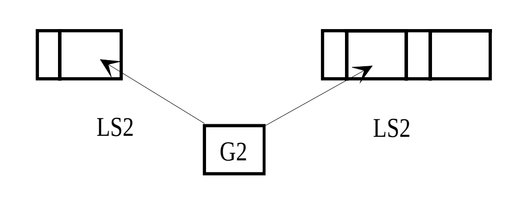
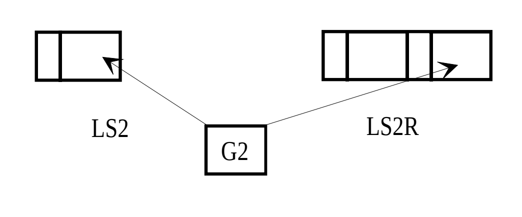
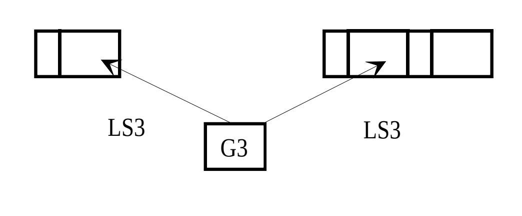
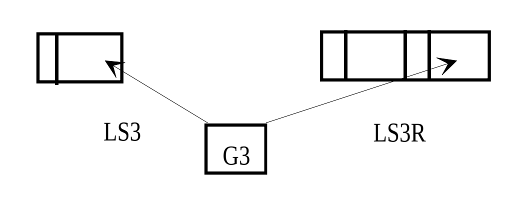
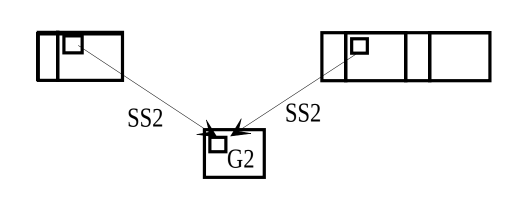
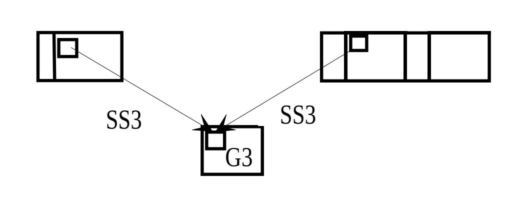
Table 7 (continued)

Facility Number	**Final Byte**	**Facilities utilised**	**7-bit code**	**8-bit code**
7	04/07	The C1 code element shall be used. Each C1 control function shall be represented by - in 7-bit code: the ESC Fe sequence; in 8-bit code: single bit combination from the CR area (columns 08 and 09).		
8	04/08	All graphic character sets comprise 94 characters.		
9	04/09	The G1, G2, and G3 graphic character sets may comprise 94 and/or 96 characters (in 7-bit and 8-bit codes).		
10	04/10	A 7-bit code is used although the environment may be able to support an 8-bit code.		
11	04/11	An 8-bit code is used.		
12	04/12	Level 1 of ECMA-43 shall be used.	See 10.3.1	
13	04/13	Level 2 of ECMA-43 shall be used.	See 10.3.1	
14	04/14	Level 3 of ECMA-43 shall be used.	See 10.3.1	
16	05/00	In addition to any other code element G0 to G3 which may be used, G0 shall be used. It will be invoked by - in 7-bit code: SI; in 8-bit code: LS0. (*)		
18	05/02	In addition to any other code element G0 to G3 which may be used, G1 shall be used. It will be invoked by - in 7-bit code: SO; in 8-bit code: LS1. (*)		

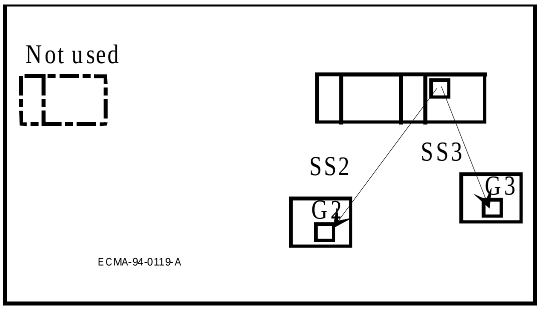
NOTE 52

Facility numbers 15 and 17 are reserved for future standardization.

Table 7 (continued)

Facility Number	Final Byte	Facilities utilised	7-bit code	8-bit code
19	05/03	In addition to any other code element G0 to G3 which may be used, G1 shall be used. It will be invoked by - in 7-bit code: SO; in 8-bit code: LS1R. (*)		
20	05/04	In addition to any other code element G0 to G3 which may be used, G2 shall be used. It will be invoked by - in 7-bit code: LS2; in 8-bit code: LS2. (*)		
21	05/05	In addition to any other code element G0 to G3 which may be used, G2 shall be used. It will be invoked by - in 7-bit code: LS2; in 8-bit code: LS2R. (*)		
22	05/06	In addition to any other code element G0 to G3 which may be used, G3 shall be used. It will be invoked by - in 7-bit code: LS3; in 8-bit code: LS3. (*)		
23	05/07	In addition to any other code element G0 to G3 which may be used, G3 shall be used. It will be invoked by - in 7-bit code: LS3; in 8-bit code: LS3R. (*)		
26	05/10	In addition to any other code element G0 to G3 which may be used, G2 shall be used. SS2 shall invoke a single character of this set in both 7-bit and 8-bit codes. (*)		
27	05/11	In addition to any other code element G0 to G3 which may be used, G3 shall be used. SS3 shall invoke a single character of this set in both 7-bit and 8-bit codes. (*)		

28 05/12 In an 8-bit code any single character invoked by SS2 or SS3 shall be represented by bit combination(s) from columns 10 to 15.



NOTE 53

Facility numbers 24, 25 and 29 to 62 are reserved for future standardization.

15.3 Data Delimiter for this Coding Method (CMD)

15.3.1 Purpose

A control function is provided to act as an instruction to return from the coding method defined in this Standard to the coding method used in an outer level of the data structure, such as an application data syntax. It may be used to indicate the end of a CC-data-element where this cannot be determined in some other way.

NOTE 54

An application which uses a data syntax in which not all the data elements are CC-data-elements in accordance with this Standard, or uses another general level of control, generally indicates the beginning of a string of character-coded information by means of the conventions of that data syntax. The end of the string may be indicated by a delimiter.

NOTE 55

The escape sequence representing this control function may be suitable for use by coding systems other than that of this Standard.

15.3.2 Specification

Name: CODING METHOD DELIMITER Acronym: CMD Coded representation: ESC 06/04

The control function CMD shall delimit (i.e. terminate) a string of data coded according to this Standard, and shall switch to a general level of control.

The use of this function is not mandatory if the outer level of data structure defines means of delimiting the string, for instance by specifying the length of the string.

15.4 Designation of Other Coding Systems (DOCS)

15.4.1 Purpose

A code-identification function is provided to designate and invoke an identified coding system different from that of this Standard, not necessarily a character code. It provides a means for switching between coding systems according to this Standard and other coding systems when it is not performed at an outer level (e.g. 15.3).

Each such other coding system is registered in the ISO International Register of Coded Character Sets (see annex B), together with a Final Byte (and where necessary one or more associated Intermediate bytes) to identify it.

This function is also recommended for use by such other coding systems for returning to the coding system of this Standard, by the use of a reserved Final Byte which identifies it.

NOTE 56

Other standards specify alternative methods for achieving an effect similar to that of DOCS, e.g. those based on the upper layers defined in ISO 7498 - Open Systems Interconnection, Basic Reference Model.

15.4.2 Specification

Name: DESIGNATE OTHER CODING SYSTEM Acronym: DOCS
Coded representation: ESC 02/05 F or ESC 02/05 I F

DOCS shall designate and invoke an identified coding system. In the coded representation of DOCS the F byte (and any associated I bytes) shall identify the coding system.

DOCS with Final Byte 04/00, without I byte, shall designate the coding system of this Standard. It is intended for use by other coding systems for returning to this coding system. It shall restore the state of the coding system to that at the time of invocation of the other coding system, that is the state established by announcer functions, and the designation and invocation state of graphic character and control character sets. Whether or not other states, e.g. the active position, are restored is outside the scope of this Standard.

DOCS with I byte 02/15 shall mean that the other coding system does not use DOCS (F = 04/00), coded as specified here, to return (it may have an alternate means to return or none at all). It shall also mean that after such a return (if any) the previous state of this coding system (i.e. announcements, designations, and invocations) is undefined.

DOCS with any other I byte, or with no I byte, shall mean that the other coding system uses DOCS (F = 04/00) to return.

Annex A

(normative)

External references to character repertoires and their coding

A.1 Methods of reference to character repertoires and their coding

Within programming languages and other methods for defining the syntax of data objects there is commonly a need to declare specified character codes that are in accordance with International Standard ISO/IEC 2022 (see 10.2).

For any fully specified code that is in accordance with International Standard ISO/IEC 2022 a precise declaration of that code should identify the following features:

– the reference number of the International Standard (i.e. ISO/IEC 2022),

– the code structure and shift functions, identified by the number of bits in a byte and the implementation level,

– the component character sets and any further single additional control functions, identified by one or more registration numbers,

– the initial state of designations and shift status.

A.2 Identification of ASN.1 character abstract syntaxes

One of the methods now in common use for defining the syntax of data objects is Abstract Syntax Notation 1 (ASN.1) specified in ISO 8824. The corresponding coded representations are specified in ISO 8825. When this method is used the forms of the references to character repertoires and coding are as indicated in this clause and in A.3 respectively.

The set of all character strings that can be formed from the characters of an identified repertoire in accordance with this Standard is defined to be a "character abstract syntax" in the terminology of ISO 8824. For each such character abstract syntax a corresponding object identifier value is defined to permit references to be made to that syntax when the ASN.1 notation is used.

A.2.1 Object identifiers

ISO 8824 annex B specifies the form of object identifier values for objects that are specified in an ISO standard. In such an object identifier the features and options of International Standard ISO/IEC 2022 shall be identified by means of numbers (arcs) which follow (i.e. are below) the arc "2022" which identifies the International Standard.

The sequence of such arcs shall be:

abstract-syntaxes (1) *reg1 reg2 reg3* ...[revisions(0) *rev1 rev2 rev3* ....]

where the set of arcs shown between [and] is optional.

The first arc, which has the value (1), identifies the object as an abstract-syntax. It shall be followed by one or a sequence of further arcs, each of which shall be a register entry number from the ISO International Register of Coded Character Sets, in ascending numerical order. This sequence shall identify all the sets of graphic characters and control functions that form a part of the code, and is notated here as *reg1 reg2 reg3* ....

The above sequence of arcs may optionally be followed by a further sequence of arcs, the first of which is the arc "revisions(0)"; the other arcs in the sequence shall be the set of revision numbers (if any) that are applicable to the set of register entry numbers identified previously. The arcs for the revision numbers shall appear in the same order as the arcs for the register entry numbers to which they correspond; where no revision number is applicable the corresponding arc shall have the value (0). Any consecutive zeroes at the end of the sequence may be omitted. The sequence is notated here as *rev1 rev2 rev3* ...

For a code which permits redesignation of character sets within a single CC-data-element (i.e. a code at Level 4 or 4C) the sequence of arcs shall identify all the sets which may be designated at points within the CC-data-element, as well as the designations that apply initially.

NOTE A.1

As an example, the object identifier for the repertoire of ISO 8859-1 is:
{iso standard 2022 abstract-syntaxes (1) 6 100}

A.2.2 Object descriptors

ISO 8824 also specifies object descriptors corresponding to object identifier values. For the above sequence of arcs the corresponding object descriptor shall be as follows:

"ISO 2022 registrations *reg1*[*/rev1*], *reg2*[*/rev2*], *reg3*[*/rev3*], ... "

where the text between pairs of brackets [and] is optional.

NOTE A.2

All spaces shall be single spaces.

A.3 Identification of ASN.1 character transfer syntaxes

The coding method for character strings that can be formed from the characters in accordance with this Standard is defined to be a "character transfer syntax" in the terminology of ISO 8825. For each such character transfer syntax a corresponding object identifier value is defined to permit references to be made to that syntax when the ASN.1 notation is used.

A.3.1 Object identifiers

In an object identifier in accordance with ISO 8824 annex B, the coded representation of the character sets of a code conforming to International Standard ISO/IEC 2022 shall be identified by means of numbers (arcs) which follow (i.e. are below) the arc "2022" which identifies the International Standard.

The sequence of such arcs shall be:

transfer-syntaxes (0) *code-level g0 g1 g2 g3 c0 c1 gleft gright*

The first arc, which has the value (0), identifies the object as a transfer-syntax. It shall be followed by one or a sequence of further arcs as indicated by the notation above.

The arc *code-level* shall identify the code structure, by reference to the number of bits in the byte and the implementation level. It shall have one of the following values:

7bit-level1 (1)	8bit-level1 (11)	8bit-level1c (21)
7bit-level2 (2)	8bit-level2 (12)	8bit-level2c (22)
7bit-level3 (3)	8bit-level3 (13)	8bit-level3c (23)
7bit-level4 (4)	8bit-level4 (14)	8bit-level4c (24)
	8bit-level2a (15)	8bit-level2ac (25)
	8bit-level3a (16)	8bit-level3ac (26)
	8bit-level4a (17)	8bit-level4ac (27)

The arcs *g0 g1 g2 g3 c0* and *c1* shall each be either a register entry number from the ISO International Register of Coded Character Sets or zero (0). Non-zero arcs *g0, g1, g2,* and *g3* shall be the register entry numbers for the sets that are designated respectively as the G0, G1, G2, and G3 code elements of the code. Non-zero arcs *c0* and *c1* shall be the register entry numbers for the sets that are designated respectively as the C0 and C1 code elements of the code. A zero arc shall indicate that no character set has been designated as the corresponding code element. For a code at Level 4 or 4C the arcs identify the initial designations only, i.e. the designations that apply at the beginning of each CC-data-element.

The arc *gleft* shall identify the code element G0, G1, G2, or G3 that initially has GL shift status in an 8-bit code, or that initially has shift status in a 7-bit code. It shall have one of the values:

g0 (0), g1 (1), g2 (2), or g3 (3)

corresponding respectively to those code elements. The default value is (0).

The arc *gright* shall identify the code element G1, G2, or G3 that initially has GR shift status in an 8-bit code. It shall have one of the values:

g1 (1), g2 (2), or g3 (3)

corresponding respectively to those code elements. The default value is (1).

In the sequence of arcs the default value of *gright*, and all consecutive zeroes that precede that value, may be omitted.

NOTE A.3

Examples of references for complete codes and their corresponding object identifiers are shown below:

– *IRV of ISO/IEC 646 (includes G0 and C0 code elements only)*
  {iso standard 2022 transfer-syntaxes (0) 7bit-level1 (1) 6 0 0 0 1}

– *ISO 8859-1 (includes G0 and G1 code elements only)*
  {iso standard 2022 transfer-syntaxes (0) 8bit-level1 (11) 6 100}

A.3.2 Restrictions on code extension facilities

Within CC-data-elements conforming to a code that is identified as specified in A.3.1:

– the announcer function (ACS) shall not appear,
– designation functions and the IRR function shall only appear when the code is at Level 4 or 4C,
– the functions CMD and DOCS shall not appear.

A.3.3 Object descriptors

ISO 8824 also specifies object descriptors corresponding to object identifier values. For the above sequence of arcs the corresponding object descriptor shall be as follows:

"ISO 2022 *code/level* G0=*reg* G1=*reg* G2=*reg* G3=*reg* C0=*reg* C1=*reg* GL=*ga* GR=*gb*"

NOTE A.4

All spaces shall be single spaces.

The notation *code/level* indicates that any one of the following phrases may occur at that point:

7-bit/level-1	8-bit/level-1	8-bit/level-1C
7-bit/level-2	8-bit/level-2	8-bit/level-2C
7-bit/level-3	8-bit/level-3	8-bit/level-3C
7-bit/level-4	8-bit/level-4	8-bit/level-4C
	8-bit/level-2A	8-bit/level-2AC
	8-bit/level-3A	8-bit/level-3AC
	8-bit/level-4A	8-bit/level-4AC

In the phrases of the form xx=*reg* the notation *reg* indicates a registration number. Any such phrase may be omitted if no character set is designated as the identified code element.

In the phrase GL=*ga* the notation *ga* indicates G1, G2, or G3. The default phrase GL=G0 shall be omitted.

In the phrase GR=*gb* the notation *gb* indicates G2 or G3. The phrase shall be omitted for a 7-bit code. For an 8-bit code the default phrase GR=G1 shall be omitted.

Annex B

(informative)

The ISO International register of coded character sets to be used with escape sequences

B.1 The ISO International Register

The ISO International Register contains character sets which have been registered in accordance with procedures specified in ISO 2375. Its purpose is to identify widely used character sets and associate with each a unique register entry number to identify it, and a unique escape sequence by means of which it can be designated conveniently.

The publication of this Register promotes compatibility in international information interchange and avoids duplication of effort in developing application-oriented character sets. Registration provides an identification for a character set but implies nothing about its status; it may or may not be part of an international or a national standard, or of an application-oriented standard. However, when such a standard is issued subsequent to the registration of a register number and an escape sequence, it is appropriate for these two forms of identification to be specified in the standard.

NOTE B.1

This register also contains registrations of other coding systems and of single control functions.

B.2 New Registrations

If a recognised body wishes to register a character set, application should be made as required in ISO 2375. Any character set can be a candidate for registration as long as it satisfies the technical requirements of ISO/IEC 2022 and the formal requirements of ISO 2375. Its characteristics will determine the type of escape sequence that can be allocated to it.

B.3 The Registration Authority

The registration procedure and the maintenance of the Register is performed by an International Registration Authority. ECMA* has been designated by the Council of the International Organization for Standardization, ISO, as the Registration Authority for ISO 2375.

ECMA carries out these duties as a free service to the international data processing community. It advises applicants on the requirements to be met by applications, circulating the applications in conformance with the procedures. It allocates the register number and the escape sequence and finally registers each character set with its specific escape sequence. After each registration owners of the Register receive the corresponding additional sheets of the Register.

* ECMA
Rue du Rhône 114
CH-1204 GENEVA
Switzerland

Phone: +41 22 849.60.00
Fax: +41 22 849.60.01
Internet: helpdesk@ecma.ch

Annex C

(informative)

Main differences between the 4th edition (1985) and the 5th edition of this Standard

C.1 The standard has been completely rearranged and rewritten, to make it more convenient to use as a reference document. In particular the specification of 8-bit codes and 7-bit codes is now presented in the following sequence:

- Characters and character sets,
- The elements of 8-bit and 7-bit codes,
- Structure of 8-bit codes,
- Structure of 7-bit codes.

C.2 Names and acronyms have been assigned to the code-identification functions, such as designations and announcers. These functions are specified in a separate section of the standard following the specification of 8-bit and 7-bit codes.

C.3 Escape sequences are now regarded only as the coded representations of code-identification functions, and of some control functions. They are specified accordingly within a single clause, separate from the specification of the functions that they represent.

C.4 A new option has been added to the specification to allow for the unique coding of graphic characters within a code whenever the same character is present in more than one of the code elements G0 to G3 if required by the application (see 7.5).

C.5 The specification of an 8-bit code may require that, following the single-shift functions SS2 and SS3, the bit combination(s) representing the invoked character shall be taken from columns 10 to 15 and not from columns 02 to 07 (see 8.4).

C.6 A new set of levels of implementation has been specified for 8-bit and 7-bit codes, based on those specified in ECMA-43 (see 10.3).

C.7 The new style of conformance clause, adopted for all standards on character coding, has been introduced.

C.8 Object identifiers conforming to Abstract Syntax Notation One (ASN.1, see ISO 8824) are specified for the character repertoires and codes that may be constructed in accordance with this Standard. The specification appears in a new annex.

C.9 The wording of the definitions has been aligned with that of ECMA-43 and ECMA-48.

Annex D
(informative)

Main differences between the 5th edition (1993) and the present edition of this Standard

D.1 The requirement for unique coding of graphic characters within a code has been made an option to be chosen according to the application (7.5).

D.2 The coded representation of graphic characters invoked by SS2 and SS3 shall be taken from the same range of column numbers within any version of an 8-bit code, i.e. either from columns 02 to 07, or from columns 10 to 15, but not from both (8.4). This feature is also recognized in the identification of levels of implementation (10.3.2).

D.3 The annex on conformance of combined originating/receiving devices, taken from ISO/IEC 10646-1, has been removed.

D.4 Object identifiers according to ASN.1 have been extended to allow for revisions of entries in the ISO International Register of Coded Character Sets to be used with Escape Sequences (annex A).

D.5 The 6th edition of this ECMA Standard has been fully aligned with the 1994 edition of ISO/IEC 2022, which is based on the 5th edition of this ECMA Standard.

This Standard ECMA-35 is available free of charge from:

ECMA
114 Rue du Rhône
CH-1204 Geneva (Switzerland)

Fax: **+41 22 849.60.01**
Internet: **helpdesk@ecma.ch**

This Standard can also be downloaded as file E035.DOC.EXE or E035-PSC.EXE from ECMANEWS.