Annex A

8-Bit Character Sets

This Annex to the Guide to the Use of Character Sets in Europe provides more detailed information about 8-bit character set standards than is found in the main body of the Guide. Annex B deals in more detail with the Universal Multi-octet Coded Character Set (UCS) specified in ISO/IEC 10646-1.

The need to represent characters by bit combinations (binary numbers) is central to the storage and processing of data by computer systems and the interchange of data between such systems. This annex gives guidance on the many standards and other specifications which have been developed to address the issues that arise from this need up and until the advent of the multi-octet code structure embodied in ISO/IEC 10646-1:1993.

Table of Contents

1 More about this guide....4

2 Limitations of this guide....4

3 User Requirements....4

3.1 Language Support....4
3.2 Page and Display Formats....5
3.3 European Requirements....6

4 Introduction to character sets....6

4.1 Historical background....6
4.1.1 The first binary codes....6
4.1.1.1 The legacy of Baudot....6
4.1.1.2 Locking shifts....7
4.1.1.3 National variants....7
4.1.2 ASCII....7
4.1.2.1 A 7-bit code....7
4.1.2.2 The legacy of paper tape....7
4.1.2.3 94 characters....7
4.1.2.4 Built-in extendability....8
4.1.2.5 International adoption....8
4.1.3 The world after ASCII....8
4.1.3.1 7-bit codes in an 8-bit world....8
4.1.3.2 8-bit codes....8
4.1.3.3 Locking shifts again....9
4.1.3.4 The International Register....9
4.1.3.5 Limits on expansion....9
4.1.4 The future is 16-bit....9
4.2 Concepts and terminology....10
4.2.1 Basic principles of ISO/IEC 2022....10
4.2.2 Code tables....11
4.2.2.1 Layout and notation....11
Structure....11
4.2.2.3 Escape sequences....12
4.2.3 Code elements....12
4.2.3.1 Code elements G0, G1, G2 and G3 of graphic characters....12

4.2.3.2 Code elements C0 and C1 of control characters....13
4.2.3.3 Other control functions....14
4.2.4 Repertoire of a code....14
4.2.5 Formal definitions....15

5 Technical Guidance....15

5.1 Application Environments....16
5.1.1 Features of sequential access....16
5.1.2 Features of random access....16
5.1.3 Use of code extension techniques....17
5.1.4 Restriction to subrepertoires....17
5.2 8-Bit Character Sets – Graphic Characters....17
5.2.1 94 and 96 position character sets....18
5.2.2 Single-byte and multiple-byte character sets....18
5.2.2.1 Nesting of character sets....18
5.2.2.2 Coding of nested sets....18
5.2.2.3 Chinese, Japanese and Korean national standards....19
5.2.2.4 Variable-length coding....20
5.2.3 Combining characters....20
5.3 Control Functions....21
5.3.1 Primary sets of control functions....21
5.3.2 Supplementary sets of control functions....22
5.3.3 Escape sequences....22
5.3.3.1 General construction....22
5.3.3.2 Two-byte escape sequences....23
5.3.3.3 Escape sequences with Intermediate Bytes....23
5.3.4 Code extension....23
5.3.4.1 Locking shifts....23
5.3.4.2 Single shifts....24
5.3.4.3 Designation of sets of control functions....24
5.3.4.4 Designation of sets of graphic characters....25
5.3.4.5 Announcement functions....25
5.3.5 Control sequences....25
5.3.6 Control strings....26
5.3.7 Control functions for text communication....27

6 Guides to standards....27

6.1 International Standards....27
6.1.1 ISO/IEC 646....27
6.1.1.1 Current edition....27
6.1.1.2 Description....27
6.1.1.3 Tutorial guidance....27
6.1.2 ISO/IEC 2022....29
6.1.2.1 Current edition....29
6.1.2.2 Description....29
6.1.2.3 Tutorial guidance....29
6.1.3 ISO 2375....29
6.1.3.1 Current edition....29
6.1.3.2 Description....29
6.1.3.3 Tutorial guidance....29
6.1.4 ISO/IEC 4873....30
6.1.4.1 Current edition....30
6.1.4.2 Description....30

6.1.4.3 Tutorial guidance....30
6.1.5 ISO/IEC 6429....30
6.1.5.1 Current edition....30
6.1.5.2 Description....30
6.1.5.3 Tutorial guidance....31
6.1.6 ISO/IEC 6937....31
6.1.6.1 Current edition....31
6.1.6.2 Description....31
6.1.6.3 Tutorial guidance....32
6.1.7 ISO/IEC 7350....32
6.1.7.1 Current edition....32
6.1.7.2 Description....32
6.1.7.3 Tutorial guidance....33
6.1.8 ISO/IEC 8859....33
6.1.8.1 Current edition....33
6.1.8.2 Description....33
6.1.8.3 Tutorial guidance....34
6.1.9 ISO/IEC 10367....35
6.1.9.1 Current edition....35
6.1.9.2 Description....35
6.1.9.3 Tutorial guidance....35
6.1.10 ISO/IEC 10538....37
6.1.10.1 Current edition....37
6.1.10.2 Description....37
6.1.10.3 Tutorial guidance....37
6.1.11 ISO/IEC 10646....38
6.1.11.1 Current edition....38
6.1.11.2 Description....38
6.1.11.3 Tutorial guidance....38
6.1.12 ISO/IEC ISP 12070....38
6.1.12.1 Current edition....38
6.1.12.2 Description....38
6.1.12.3 Tutorial guidance....38
6.2 International Registers....39
6.2.1 ISO 2375 Register (International Register of Coded Character Sets to be used with Escape Sequences)....39
6.2.1.1 Current edition....39
6.2.1.2 Description....39
6.2.1.3 Tutorial guidance....39
6.3 European Standards....41
6.3.1 EN 1922....41
6.3.1.1 Current edition....41
6.3.1.2 Description....41
6.3.1.3 Tutorial guidance....41
6.3.2 EN 1923....41
6.3.2.1 Current edition....41
6.3.2.2 Description....41
6.3.2.3 Tutorial guidance....42

1 More about this annex

The requirement for compatibility between newer and older equipment has led to the standards of the present day containing legacies from decisions taken many years ago. The reasons behind those decisions are often no longer relevant and their present day legacies may appear merely as unnecessary oddities and complexities. This annex includes some historical background which however is not necessary for an understanding of the remainder of the text.

As work on character sets has developed, there has been a gradual refinement of the concepts involved. This has led to character set standards and other literature making use of technical terms that can be a barrier to the reader. It may be helpful to read section 1.2, Concepts and terminology, before exploring the remaining sections in detail.

This gradual evolution of character set standards has led to technical innovations designed to increase the capabilites of coded character sets while remaining backwardly compatible with what has gone before. Within this evolved framework it is now possible to support a wide range of languages. The wider the range that it is required to support simultaneously, however, the more complex is the technical innovation required. For further information see section 3.1, Language support.

Not all the technical innovations are compatible with all the ways that character data may be used by applications. Section 5.1, Application Environments, provides guidance on these limitations.

Other sections provide greater detail on particular issues.

2 Limitations of this annex

This annex does not cover, or only briefly covers, the following topics:

- The new multiple-octet coded character set standard ISO/IEC 10646;
- The use of coded character sets in Open Systems Interconnection (OSI) standards;
- The presentation of character data on display devices or as printed images;
- Transformations between different coding methods;
- Sorting (collating) of coded character set data.

3 User Requirements

The ultimate user of equipment that makes use of coded character sets is concerned with such matters as which languages it supports, whether page layout information can be preserved during interchange of documents, and other matters of a similar nature. This section is concerned with the facilities available in character set standards to meet such requirements.

3.1 Language Support

One of the prime requirements in the use of character sets is to be able to support the languages of concern to the user. A number of different International Standards have been developed to provide multilingual support. In addition, other languages are supported by the character sets of the International Register of Coded Character Sets to be used with Escape Sequences.

The following is an index to the various sections containing information on the languages supported by such standards and register entries.

- For variants of the ASCII 7-bit character set designed primarily to support one individual language in a Latin script, see 6.1.1, ISO/IEC 646. See the historical introduction in 4.1 to find out more about the origins of ASCII.

- For multilingual support obtained by supplementing ASCII to create an 8-bit code, see 6.1.8, ISO/IEC 8859. This standard permits simultaneous support either for a range of languages in the Latin script, or for basic Latin letters (A-Z and a-z, but not þ, ø, etc. or accented letters) together with an alphabet in any one of the Greek, Cyrillic and other scripts.

- For the greatest support of languages in the Latin script that can be achieved with a single 8-bit code, see 6.1.6, ISO/IEC 6937. This is achieved by the use of non-spacing diacritical marks, so permitting more characters to be represented than there are positions in the code table. This complication has its price; see the guidance on application environments in 5.1.

- To achieve the level of support of Latin languages that is provided by ISO/IEC 6937 but without the use of non-spacing diacritical marks, see 6.1.9, ISO/IEC 10367. This is achieved by the use of locking shifts. This complication also has its price; again see the guidance on application environments in 5.1.

- To support up to three of the four scripts Greek, Cyrillic, Hebrew and Arabic simultaneously with basic use of the Latin alphabet, or up to two of them simultaneously with a wide range of Latin languages, see 6.1.9, ISO/IEC 10367. This is achieved by the use of both locking shifts and non-spacing diacritical marks. The provisos mentioned in both the preceding entries in this list then apply.

- To support Chinese, Japanese and Korean ideographic scripts, see 6.2.1 on the International Register.

It may be helpful to read the introduction to concepts and terminology in 4.2 before following some of the above references.

3.2 Page and Display Formats

Text that is communicated by the use of coded character sets is usually intended ultimately for presentation on a screen or printed page. There is a need to be able to communicate, with that data, information concerning the layout of the text on the screen or page.

Such layout information may be either

- embedded in the communicated text, or

- separated from the text as elements of some communication protocol.

Two standards are available that provide coded control functions for embedding layout information in communicated text.

- Control functions applicable to character-imaging devices in general are specified in ISO/IEC 6429; see 6.1.5. This standard includes facilities that permit texts in different scripts to he presented in opposite directions, such as mixed Latin and Arabic, or Latin and Hebrew.

- Control functions intended specifically for the control of page layout are specified in ISO/IEC 10538; see 6.1.10. This standard includes facilities both for fixed format and for automatically reformattable text. The former assumes that both sender and receiver have the same fonts available. The latter permits reformatting where sender and receiver have access to different fonts.

3.3 European Requirements

There are various published sources of guidance on the use of character set standards in Europe. These include:

- Guidance on European character repertoires for use with 8-bit single-byte coding, given in EN 1923; see 6.3.2.
- Guidance on the interchange of character data by means of the Telex network in Europe, given in EN 1922; see 6.3.1.
- Guidance on the use of character sets in Europe in connection with OSI Abstract Syntax Notation One (ASN.1), given in ISO/IEC ISP 12070; see 6.1.12.

4 Introduction to character sets

Formal specifications for coded character sets are often couched in a language of escape sequences, ISO-IR numbers, CL and GL areas, C0 and G0 code elements and more. There are 7-bit and 8-bit codes and subtle distinctions between 94-character and 96-character sets. This annex cannot escape from the use of such language but this introduction provides an explanation for the novice.

4.1 Historical background

The assignment of specific bit combinations to a particular set of characters constitutes a coded character set, or more concisely, a code. The larger the set of characters, the greater the number of bits required for the coding. Any increase in the number of bits causes a corresponding increase in cost in the systems that use the resulting code. This need not be a monetary cost, it may instead be a cost in terms of resources, but it is nevertheless real. The historical development of coded character sets is a story of balancing the desire for more characters against that for the fewest possible number of bits. The decisions that were made have had consequences that have lasted long after the pressures that led to them have eased. This section gives some account of that history.

4.1.1 The first binary codes

4.1.1.1 *The legacy of Baudot*

The first binary coded character set was a 5-bit code patented by Jean-Maurice-Émile Baudot (1845-1903) in 1874 in connection with his invention of a precursor of the teleprinter. Since the device was operated by electromechanical means, even one further bit would have added significantly to the complexity of the equipment. In 1932 the CCITT (Comité Consultatif International Télégraphique et Téléphonique) standardized a 5-bit code for teleprinters, based on that of Baudot, which is the code of the international telex (teleprinter) network to the present day. This is known as the International Telegraphic Alphabet No.2, also as CCITT code No.2 or simply as the Baudot code. It was last re-issued as ITU-T Recommendation S.1 (1993).

4.1.1.2 *Locking shifts*

A 5-bit code has room for 32 characters, which is not enough even for the 26 letters A to Z and the ten digits. To get round this, a teleprinter operated by the Baudot code has a shift lock in the manner of a typewriter. This locks 26 “keys”, i.e. bit combinations, into one of two modes. In the alphabetic mode they print the letters A to Z. In the numeric mode some of these bit combinations print the ten digits 0 to 9 and various punctuation marks while the remainder operate certain functions such as line feed, carriage return and sounding a bell. The effect of the remaining 6 bit combinations is not affected by the shift lock. Two of these six are used to switch the shift lock between the two modes. In this way the 5-bit code conveys 58 different signals (26 times 2, plus 6).

4.1.1.3 *National variants*

Right from the beginning it was recognised that different countries had different needs. Although 58 character positions does not give much room for flexibility, the 1932 standard for the Baudot code filled only 55 of the available positions. The remaining three character positions were then available for national use.

4.1.2 ASCII

4.1.2.1 *A 7-bit code*

In the late 1950's the American Standards Association (now the American National Standards Institute) set about the development of a new code for the communications and data processing industries. By then, there was a need for further character positions to be available, both for printing and control purposes. To avoid the need for shift operations, it was agreed to develop a 7-bit code. This has 128 bit combinations available. The code was to be known as the American Standard Code for Information Interchange, or ASCII.

4.1.2.2 *The legacy of paper tape*

By that time it was common to store data on punched paper tape, for input to data processing systems or automated communication equipment. A “1” bit was represented by a hole and a “0” bit by the absence of a hole. Since a row of 7 zero bits would be indistinguishable from blank tape, the coding “0000000” would have to represent a NULL character (absence of any effect).

Since holes, once punched, could not be erased but an erroneous character could always be converted into “1111111”, this bit pattern was adopted as a DELETE character. When received, or otherwise processed, it again was to have no effect but it could be punched on top of any other character to erase that character's effects.

4.1.2.3 *94 characters*

A design decision was taken to reserve the first 32 bit combinations (i.e. with the two most significant bits being 0) for control characters. This range includes the NULL character but not the DELETE character, so it leaves 95 bit combinations for printing characters. The printing characters (including SPACE) were to be arranged in an order that could be used for sorting purposes. The SPACE character is normally sorted before any other printing character and so was allocated the first position among the printing characters. There are then 94 contiguous positions for printing characters between “1100000” (SPACE) and “1111111” (DELETE).

This division of the code positions into 32 for control starting with NULL, followed by 94 printing characters lying between SPACE and DELETE, has dominated the structure of coded character sets right to the present day; see 4.1.4 below.

The first ASCII standard was published in 1963, but at that time it left many bit combinations unallocated. It included capital letters but not small letters. The ASCII standard as we know it today dates from 1968.

4.1.2.4 *Built-in extendability*

Although the use of a 7-bit code for ASCII was designed to avoid the use of the locking shifts of the Baudot code, a pair of locking shift codes SHIFT IN (SI) and SHIFT OUT (SO) were included in the set of control characters to allow for future extension. An ESCAPE character was also included, to act as a non-locking route to extension.

4.1.2.5 *International adoption*

The later stages of the ASCII story were joint developments with the ISO subcommittee ISO/TC97/SC2. This led to the publication in 1967 of ISO Recommendation 646 (ISO had Recommendations rather than Standards at that time). Just as with the Baudot code, the need for national variants was recognised. With the greater freedom offered by 94 printing characters, 10 positions were reserved for national use. To ensure maximum consistency when these 10 positions were not all required, the recommendation provided a default assignment of characters to these positions. The version in which all the default assignments were used was known as the International Reference Version (IRV).

The most recent edition of this standard is the third edition, ISO/IEC 646:1991, which superseded the second edition of 1983. In these editions there are still 10 positions for national use and in addition a further two that have two alternative graphics assigned (number sign versus pound sign, dollar sign versus a generic currency sign). Both these and the 10 national use positions have specific assignments in the IRV. However, it is important to be aware that the IRV of ISO/IEC 646 was changed between the 1983 and 1991 editions. To conform with *de facto* usage, the 1991 edition recognised ASCII as the new IRV. The IRV of the 1983 edition specified the generic currency sign alternative in the choice between that and the dollar sign.

4.1.3 The world after ASCII

4.1.3.1 *7-bit codes in an 8-bit world*

The 7-bit codes that followed ASCII, such as for other scripts (Greek, Arabic, etc.), followed the basic structure of ASCII. They kept the 32 control characters, SPACE and DELETE and changed only the remaining 94 printing characters. When 7-bit codes were used in the 8-bit environment provided by most computers, the most significant bit was set to “0”. This leaves the NULL character being “00000000” which is consistent with its original design intention but it codes the DELETE character as “01111111”, so no longer having all “1” bits.

4.1.3.2 *8-bit codes*

This approach led to a natural method of extension to accomodate more characters: use a second 94-character 7-bit code and distinguish it by setting the most significant bit to “1”. Such an extended code can be transmitted through 7-bit communication channels by use of the SI and SO locking shifts of ASCII. There is no need for second SPACE and DELETE characters, so these positions are unassigned in the second code. When 7-bit codes came to be designed specifically for use in this extended area, they could use the full 96 printing positions.

This design for 8-bit codes has some immediate consequences. Viewed in binary sequence the control characters are no longer contiguous; there are two control areas “000xxxxx” and “100xxxxx” and two separated areas for printing characters, a lower area with the most significant bit set to “0” and an upper area with it set to “1”. Between them they can accommodate 190 printing characters (94 plus 96) excluding SPACE.

4.1.3.3 *Locking shifts again*

The 8-bit codes described above have room for, say, accented letters or Greek letters in the upper half, but not both. The need to cater for both at once gave rise to an obvious further extension: locking shifts for 8-bit codes. As with any locking shift mechanism, it is only suitable for communication rather than for data processing, but this route was followed. A second set of two 7-bit codes could then be accommodated, to be shifted as required into the lower and upper areas for printing characters. This gives a total of four 7-bit codes in use simultaneously. There is no need to restrict usage to one alternative code for the lower area and another for the upper, so mechanisms were set up

to shift (invoke) any of the four 7-bit codes into the lower area and, independently, any other into the upper area.

4.1.3.4 *The International Register*

The limit of four 7-bit codes is in fact arbitrary; it would be possible to have even more 7-bit codes "on standby" and even more shift mechanisms for invoking them. But four is enough for most needs, and once this is exceeded there seems no particular reason to stop at any other number. The next stage in code extension was therefore to permit the choice of the four 7-bit codes to be changed by means of control functions. This is like sending a telephone message to a user of an interchangeable typehead ("golfball") typewriter to change the typehead.

This is more difficult as there has to be some method of identifying the new choice of code to the remote party to the communication. In effect there has to be a catalogue of typeheads from which one can choose, with catalogue numbers that one can communicate. This was achieved by means of an International Register of such codes, established by ISO. Each registered code is assigned a number and is referenced as ISO-IR xx. In this register, for example, the IRV of ISO 646:1983 is ISO-IR 2 and that of ISO/IEC 646:1991 (ASCII) is ISO-IR 6. This register enables new codes to be selected (designated) and subsequently shifted into use (invoked), all through the use of control functions.

The structure of character codes and their extension techniques that is described above is formalised in the international standard ISO/IEC 2022. The International Register is maintained according to procedures laid down in ISO 2375 and is published by Japanese standards institution (JISC) under the authority of ISO. The register may be accessed on-line. It currently contains over 200 registered coded character sets.

4.1.3.5 *Limits on expansion*

Not all implementations of communications protocols may be able to cope with all the possibilities of this complex system of code extension by designation and invocation. There needs to be a way of notifying the remote user of the intention to use only a selection of the available facilities. This was achieved by means of further control functions, known as announcers. These make it possible, for example, for a system to announce that it will only use a 7-bit code (or an 8-bit code) with no code extension facilities.

The summary of this section on the world after ASCII has skipped over a number of difficulties that arise in these code extension techniques. In particular, attention has been concentrated on the printing characters. The control characters also have their extension problems. An account with greater precision is given in section 4.2.

4.1.4 The future is 16-bit

With the growing processing power of computers and the increasing bandwidth of communications channels, the pressure to squeeze an ever increasing number of characters into an 8-bit code structure has diminished. A need has arisen for a simpler structure at the expense of more bits. This need has given rise to a complete rethinking of code structure for a world of 16-bit and even 32-bit processing and communication. From it has risen a new international standard, ISO/IEC 10646, the Universal Multiple-Octet Coded Character Set.

It is interesting to note that even this "ultimate" standard retains some past legacies. Control functions are coded according to ISO/IEC 2022, although the code extension functions of that standard are forbidden. The first 32 bit combinations are therefore reserved for control purposes. The next 95 bit combinations contain the printing characters of ASCII including SPACE. This brings one to the bit combination "00...001111111" (the dots denote enough zeroes to fill either 16 or 32 bits, as the case may be). The legacy of paper tape survives. This is still reserved for the DELETE character!

It is the intention that ISO/IEC 10646 will be, in some sense, the last character set standard. It is planned as a multi-part standard, of which part 1 was published in 1993. Future parts will add to the code, and since it has the potential to fill a 32-bit code space, it has the capacity to be extended to meet all foreseeable future needs. It has the ultimate aim of including all characters that have ever been used for communication. The coding of ancient runes has already been standardized, that of Egyptian hieroglyphics is for future study.

More detailed information may be found in Annex B, which deals specifically with the UCS code structure.

4.2 Concepts and terminology

Many of the 7-bit and 8-bit coded character sets in use today share a common structure. This structure, together with notation and terminology for referring to its various elements, is laid down in ISO/IEC 2022. Some familiarity with the main features of that structure is needed to read this annex. These are summarised in this section.

The Universal Multiple-Octet Coded Character Set (UCS) that has been developed more recently, specified in ISO/IEC 10646-1, lies outside the ISO/IEC 2022 structure. Detailed guidance on the structure of the UCS is found in Annex B.

4.2.1 Basic principles of ISO/IEC 2022

The construction of a character code according to ISO/IEC 2022 is most simply explained by a mechanical analogy. It is like a typewriter that takes interchangeable typeheads (“golfballs”). A typewriter without a typehead can't actually type anything but its mechanisms are all in place. The non-printing keys, such as the space bar and backspace, still operate. It is only the printing characters that are missing.

The typehead itself is an inert object, but once placed on the typewriter then each key on the typewriter will print the character that is at a specific position on the typehead. Change the typehead and the typewriter prints different characters, but the relationship between keys and character positions does not change.

The role of the typewriter is taken in ISO/IEC 2022 by a code table. There is one code table for 7-bit codes and another for 8-bit codes. Each code table provides a linkage between character positions and bit combinations. Certain of these positions are already assigned, for the SPACE, DELETE and ESCAPE characters, but the vast majority of character positions are empty. The table is waiting for its equivalent of a typehead.

The role of the typehead is taken in ISO/IEC 2022 by a code element of graphic characters. Such a code element contains a pattern of graphic characters that can be overlaid on (part of) the empty code table. Once overlaid, it provides a graphic character at each of the overlaid positions. The combination of code table and code element completes (part of) the code; the character at a particular position is coded by the bit combination assigned to that position.

The next few paragraphs expand on this model in a more precise way.

4.2.2 Code tables

4.2.2.1 *Layout and notation*

ISO/IEC 2022 defines the structure of a code table separately for 7-bit and for 8-bit codes. A 7-bit code table consists of 128 positions arranged in 8 columns and 16 rows. An 8-bit code table consists of 256 positions arranged in 16 columns and 16 rows. The rows and columns are numbered starting

from 0, and by convention a leading zero is included where necessary to make all row and column numbers have two (decimal) digits. A diagram to illustrate the 8-bit case is given below in figure 1.

The notation *xx*/*yy*, e.g. 01/15, is used to label the table position that is in column *xx* and row *yy*. The same notation is used to identify a bit combination, with *yy* being the decimal number whose binary form consists of the least significant four bits of the bit combination and *xx* being similarly related to the most significant four (for an 8-bit code) or three (for a 7-bit code) bits. This notation provides a natural correspondence between positions in the code table and bit combinations of the code.

4.2.2.2 *Structure*

The 8-bit code table is divided into four named areas:

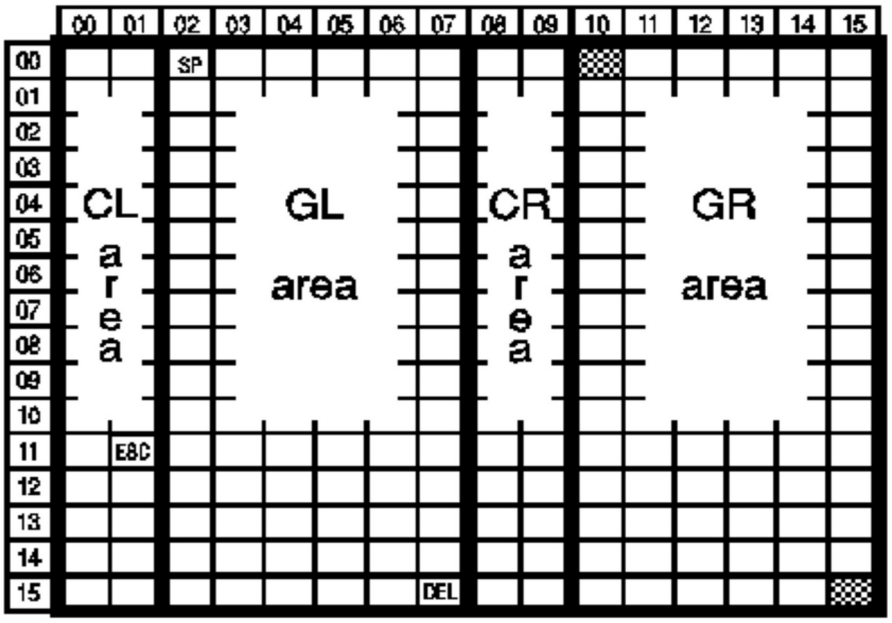


Figure 1: Structure of an 8-bit code table

- Columns 00 and 01 are called the CL area;
- Columns 02 to 07 are called the GL area;
- Columns 08 and 09 are called the CR area;
- Columns 10 to 15 are called the GR area.

The 7-bit code table is similarly divided, but it only has CL and GL areas. The 8-bit code table is illustrated in figure 1.

ISO/IEC 2022 requires that the bit combinations in the CL and CR areas shall be used to represent control functions or be left unused. Only those in the GL and GR areas may be used to represent graphic (printing) characters.

Certain characters have fixed fixed assignments in both the 7-bit and 8-bit code tables as follows:

- 01/11 is assigned to the ESCAPE character (acronym: ESC);

- 02/00 is assigned to the SPACE character (acronym: SP);
- 07/15 is assigned to the DELETE character (acronym: DEL).

These are also shown in the figure. The reasons behind the assignments for SPACE and DELETE are described in 4.1.2.

4.2.2.3 *Escape sequences*

The third of the characters with fixed assignments is the ESCAPE character. This is also in the position in which it was put during the design of ASCII. However, the reason that the ESCAPE character is so important as to require permanent assignment is more recent.

As the development of coded character sets has progressed, there has become an increasing need to be able to code control information that contains parameters. To achieve this, the concept of a control character such as CARRIAGE RETURN (CR) has given way to the more general concept of a control function. The coding of a control function is introduced by a distinctive bit combination, but it is followed by further bit combinations that pass parameter information in coded form. The syntax of these further combinations ensures that they are self-delimiting, i.e. that the end of the coding of the control function may be identified by a suitable algorithm. The ESCAPE character is one such introducer. The complete sequence of bit combinations that represents a control function coded in this manner is known as an escape sequence.

More details of the coding of escape sequences are given in section 5.3.

4.2.3 Code elements

ISO/IEC 2022 constructs a complete code from a selection of the following code elements:

- Four code elements named G0, G1, G2 and G3 containing graphic characters, arranged as possible overlays for the GL and GR areas of the 8-bit code table;
- Two code elements named C0 and C1 containing control characters, arranged as possible overlays for the CL and CR areas of the 8-bit code table;
- A number, possibly zero, of other control functions.

All of these elements may be present in either a 7-bit or an 8-bit code. Each of these types of code element will now be considered in more detail.

4.2.3.1 *Code elements G0, G1, G2 and G3 of graphic characters*

The code elements G1, G2 and G3 may each provide assignments for either 94 or 96 character positions. A set with 94 positions would provide assignments for positions 2/1 to 7/14 of the GL area or 10/1 to 15/14 of the GR area, i.e. excluding the positions assigned to SP and DEL in the GL area and the two corresponding shaded positions in the diagram above for the GR area. A set with 96 positions would provide assignments for all the 96 positions of either the GL or GR area. The code element G0 is similar but only the 94-position option is permitted.

Below is a diagram of a 94-position code element that is suitable for use as any of G0 to G3. It is in fact the ASCII character set:

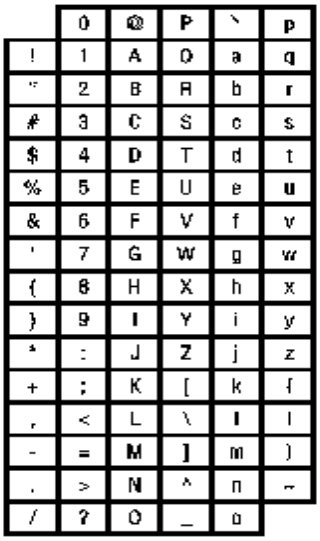


Figure 2: A 94-position code element

The process described above as overlaying is known technically, in ISO/IEC 2022 terminology, as invocation. After being invoked, the code element concerned is said to have GL or GR shift status, as the case may be. Clearly at most two of the code elements G0 to G3 may be invoked simultaneously in an 8-bit code, one in each of the GL and GR areas, and at most one in a 7-bit code where there is no GR area. The mechanism of invocation is described in more detail below.

For the code element illustrated, when it has GL shift status the character "A" is represented by the bit combination 04/01. When it has GR shift status it is represented by the bit combination 12/01.

More details of the use of the code elements of graphic characters are given elsewhere in this annex.

4.2.3.2 *Code elements C0 and C1 of control characters*

Control characters have a name and an identifying acronym, but no graphic representation. Examples of control characters are BACKSPACE (BS), BELL (BEL), START OF HEADING (SOH), SINGLE-SHIFT 2 (SS2) and ESCAPE (ESC). They are a special case of a more general concept, the control function, as explained above concerning escape sequences.

The code elements C0 and C1 each provide assignments to control characters for the 32 character positions of either the CL or CR area. If the code has a C0 code element then this is permanently invoked in the CL area. A C0 code element is required to have the ESCAPE character in position 01/11 so that its invocation does not affect the availability or coding of this control character.

If an 8-bit code has a C1 code element, it would normally be permanently invoked in the CR area. This is not possible for a 7-bit code since there is no CR area in the code table. Instead, the characters of the C1 code element are represented in a 7-bit code by means of an escape sequence. This representation is also permitted for an 8-bit code, as an alternative to invocation in the CR area. In a

particular code, only one of the two alternatives is permitted. The choice should form part of the specification of an 8-bit code.

4.2.3.3 *Other control functions*

A code with a full range of Cn and Gn code elements requires access to a substantial number of control functions. ISO/IEC 2022 makes provision for control functions to meet the following needs, and others besides:

- locking shifts for the invocation of the Gn code elements;
- single shifts to simplify the selection of isolated characters from the code elements that are not invoked;
- dynamical designation of the character sets used for the Gn and Cn code elements, to permit these code elements to be changed during the course of a single communication;
- announcement of code structure facilities.

Announcement of facilities permits the choice of particular options to be notified to the remote party to the communication, such as whether the characters of the C1 code element are to be coded in an 8-bit code by means of escape sequences or by invocation of the element to the CR area. Further information about control functions of each of the above types is given in 5.3.4.

Although some of these control functions may be coded by a single control character in either the C0 or C1 code elements, many of them require the use of an escape sequence; see 5.3.3. The set of available control functions constitutes the final element of a code specification.

The presence of the ESCAPE character in both the 7-bit and 8-bit code tables, before any other code elements are designated, permits all of the Cn and Gn code elements to be designated dynamically. Since Cn code elements do not require separate invocation, the control functions they provide are immediately available for use. These in turn can be used to invoke the Gn code elements as required. The combined effect of all available facilities is to permit a complete code specification to be communicated to a remote party if required.

4.2.4 Repertoire of a code

It is sometimes convenient to be able to refer to the set of characters that can be represented by a code, in a manner abstracted from the details of that representation. This set of characters is known as the *repertoire* of the code.

The concept of a repertoire is more subtle than it may seem to be at first. Certain character set standards permit two or more characters to be combined in specified ways to create new characters that belong to the repertoire but which are not themselves represented *in* the code. It is this distinction between representation *in*, and representation *by*, a code that causes the subtlety.

One means of combining two characters is by use of the BACKSPACE control character to superpose two images. This is still permitted in the 7-bit code of ISO/IEC 646, but not in more recent character set standards. A more recent technique is to specify that certain characters of a code are non-spacing, so that superposition may be achieved without the use of BACKSPACE. The most significant use of non-spacing characters is that of ISO/IEC 6937. Non-spacing characters are in fact only one example of the more general concept of a *combining character*.

4.2.5 Formal definitions

To ensure precision, the character set standards provide formal definitions of the terms that they use. The following extract from ISO/IEC 2022 gives the definitions of terms used above in this discussion of concepts and terminology.

bit combination: An ordered set of bits used for the representation of characters.

byte: A bit string that is operated upon as a unit. [Note that this definition permits 7-bit, 8-bit and even 16-bit bytes, although common parlance uses the term exclusively for 8 bits. Character set standards use the term "octet" when a restriction to 8 bits is intended.]

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.

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

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

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.

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

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.

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.

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.

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.

to invoke: To cause a designated set of characters to be represented by one or more bit combinations of a coded character set.

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

5 Technical Guidance

This section provides a description of the properties of character sets that may be useful, or which should be avoided, in various application environments. It also provides a more detailed account of the construction of graphic character sets and control functions than is provided in section 4 in the Introduction.

The chapter on graphic characters describes the subtle differences in use between character sets with 94 and 96 positions. It explains how both types of set can make use of either single-byte or multiple-byte coding, the latter providing access to the large character sets required for Chinese, Japanese and

Korean ideographic scripts. It also explains how, even without the use of multiple-byte coding, use of combining characters permits a coded character set to represent more characters than it has code positions.

The chapter on control functions describes the two main sources of such functions:

- the ISO 2375 Register, more properly known as the *International Register of Coded Character Sets to be used with Escape Sequences*, and
- the large repertoire of additional control functions that has been standardized in ISO/IEC 6429.

It explains the use of such control functions, with particular emphasis on those control functions that are used for code extension.

5.1 Application Environments

One of the most important distinctions between different application environments is whether the coded data is required for sequential access or for random access. Both forms of access may occur within a single application, e.g. data may be read sequentially from a storage medium into random access memory. Normally the encoded binary data would be read directly from the storage medium to the random access memory in such circumstances. It is possible, however, to transform the data from one character code to another during the transfer process. If this facility is available, different codes may be chosen which optimise the benefits for each form of access.

5.1.1 Features of sequential access

Sequential access permits the use of control functions that change the mapping between bit combinations and characters. This is known as code extension. The simplest such control function is the use of a locking shift mechanism (as on a typewriter) to switch between two such mappings. Use of locking shifts dates back to the earliest teleprinter codes; see section 4.1 for more information. Modern code extension techniques permit both locking shifts and single shifts to be used, the latter affecting only the immediately following character.

When a 16-bit, or even a 32-bit, code may be used then there is no need for such code extension techniques. When there are reasons (such as compatibility with existing equipment) why an 8-bit, or even a 7-bit, code is to be used then user requirements concerning the character repertoire may compel the use of such techniques.

5.1.2 Features of random access

Random access requires each unit of data to be complete in itself, so that it can be interpreted without reference to anything that may precede or follow it. It normally also requires that the boundaries between units of data must be fixed. For example, in byte-oriented storage of data with a code that uses two bytes per character, it may be required that each character code starts at an even address. No such algorithm is possible if the code uses a variable number of bytes in the representation of its characters. However, it may be acceptable to use such a code if, for example, examination of a fixed number of bytes at any point in the data permits the boundaries of the character representations to be determined. This property will here be called auto-resynchronization. Whether or not this is acceptable will depend on the application concerned.

The need for each unit of data to be complete in itself prevents the use of code extension by means of locking shifts. A code which is extended by means only of single shifts is, in effect, a code that uses a variable number of bytes. The coded representation of the single-shift control function may be considered as part of the representation of the character. Such a code is also auto-resynchronizing

provided that the coded representation of the single-shift function is a single byte that cannot occur in the data stream for any other reason.

5.1.3 Use of code extension techniques

A comprehensive set of techniques for code extension with 7-bit and 8-bit codes is given in ISO/IEC 2022. An introduction to these facilities is given in section 4.2.

These code extension techniques permit up to four 7-bit codes to be selected and then brought into use by means of shift mechanisms. For use with a 7-bit code, only one may be shifted into use at any time but for use with an 8-bit code, two may be brought into use simultaneously. There are mechanisms for communicating between the users which 7-bit codes have been selected, and even for changing this selection during the flow of data. Various levels of implementation are defined, each permitting the use of only a specified selection of the code extension techniques.

For 8-bit codes, greater consistency in the use of code extension techniques may be obtained by requiring conformance to ISO/IEC 4873; see 6.1.4. This fixes the left-hand half of the code table permanently as the ASCII character set and restricts the right-hand half to be a single-byte code. It therefore excludes the 7-bit two-byte codes for Chinese, Japanese and Korean that are permitted under ISO/IEC 2022 itself (more information about these codes is given in 5.2.2.3). It still permits the selection of any three such 7-bit codes for mapping by shift mechanisms into the right-hand half of the table (one of the four 7-bit sets of ISO/IEC 2022 is now permanently the ASCII set). It again specifies various levels of implementation.

Even greater consistency, at the expense of even less flexibility, may be obtained by requiring conformance to ISO/IEC 10367; see 6.1.9. This requires the three 7-bit codes of ISO/IEC 4873 to be chosen from 12 such codes that are specified in the standard.

5.1.4 Restriction to subrepertoires

There is, of course, nothing that compels any user of a coded character set to make use of all the characters that can be represented by that set. However, a recipient of coded data will not normally know that the originator of that data was not going to use all these characters. For many purposes this is unimportant. But if it is required to change the coding to that of a different coded character set, it may be desirable to know that the data will only contain characters that are contained in both repertoires.

The control functions specified in ISO/IEC 6429 include one known as IDENTIFY GRAPHIC SUBREPERTOIRE (IGS). This is provided solely for the purpose of indicating that data coded in accordance with ISO/IEC 10367 is in fact being restricted to a subrepertoire of the full repertoire of that standard. The subrepertoire concerned is identified by its number in an International Register. The manner in which this is coded is described in 5.3.5. Procedures for the registration of subrepertoires of ISO/IEC 10367 are laid down in ISO/IEC 7350; see 6.1.7.

5.2 Graphic Characters

Section 4.2 introduced the four code elements G0, G1, G2 and G3 and explained how these each make available 94 or 96 positions for the allocation of characters. This section explains the facilities available through use of these code elements. It shows in particular how they may provide for the representation of more characters than there are code positions.

5.2.1 94 and 96 position character sets

ISO/IEC 2022 provides for sets of graphic characters to make use of either 94 or 96 code positions. It also prohibits the characters SPACE and DELETE from being assigned in any such set. When these sets are invoked,

1. a 94 position set in the GL area provides assignments for bit combinations 02/01 to 7/14;
2. a 94 position set in the GR area provides assignments for bit combinations 10/01 to 15/14;
3. a 96 position set in the GL area provides assignments for bit combinations 02/00 to 7/15;
4. a 96 position set in the GR area provides assignments for bit combinations 10/00 to 15/15.

All four possibilities are permitted. When a 96 position set is invoked in the GL area it overlays the positions 02/00 and 07/15 that are otherwise assigned to the SPACE and DELETE characters. The characters SPACE and DELETE are therefore not available in this situation. When a 94 position set is invoked in the GR area, the bit combinations 10/00 and 15/15 shall not be used.

ISO/IEC 2022 permits the G1, G2 and G3 code elements to be sets of either 94 or 96 positions but the G0 set is required to have only 94 positions. It also permits the G1, G2 and G3 code elements to be invoked into either the GL area or the GR area of the code table but the G0 code element is only permitted to be invoked into the GL area.

5.2.2 Single-byte and multiple-byte character sets

5.2.2.1 *Nesting of character sets*

ISO/IEC 2022 provides for two alternative types of allocation to the code positions of a 94 or 96 position set:

1. Each position may either have a character assigned to it or be left unused, or
2. Each position may either have a further 94 or 96 position set assigned to it or be left unused.

In the second case a 94 position set may only have its positions allocated to further 94 position sets, and similarly a 96 position set may only have its positions allocated to further 96 position sets. Nesting of sets within sets is permitted to any depth.

5.2.2.2 *Coding of nested sets*

When a nested set is invoked, more than one bit combination (byte) is required to represent an individual character. A sequence of bytes is used that may be processed by the following algorithm:

1. Take the next byte in the sequence (which initially will be the first byte). It identifies either a character or a character set at the code position referenced by that byte in the currently invoked set. If it identifies a character set, go to step 2. If it identifies a character, go to step 3.
2. The identified character set is invoked, for processing the next byte only, into the same area (GL or GR) of the code table as the set currently being processed, therefore replacing it. Processing is then repeated from step 1.
3. The identified character is the character represented by the byte sequence. Processing is complete.

The effect of this algorithm is that the characters of a nested set may be represented by a sequence of one or more bytes with the following properties:

- every character of the set is represented by the same number of bytes;
- every byte of the sequence is in the range appropriate to the status of the set as a 94 or 96 position set invoked into the GL or GR area as described above.

A character set that is nested in this way is called a *multiple-byte set*. A set that is not so nested is called a *single-byte set.*

As an illustration of the effect of the coding algorithm, if a character would be represented by the sequence 03/01 05/04 when a particular two-byte set is invoked in the GL area then it would be represented by 11/01 13/04 if the same set were invoked into the GR area.

5.2.2.3 *Chinese, Japanese and Korean national standards*

Two-byte coded character sets have been registered in the ISO 2375 Register to permit Japanese, Chinese and Korean ideographic scripts to be coded within the ISO/IEC 2022 code structure. These sets are taken from corresponding national standards. They are in fact very comprehensive character sets that provide multilingual facilities; they are not confined to the ideographic characters of the languages concerned. Particular examples are as follows:

- ISO-IR 87 (ESC 02/04 xx 04/02) : Japanese standard JIS C 6226:1983

  This 94-position two-byte set contains 6877 graphic characters that include 147 symbols, digits 0-9, Latin letters A-Z and a-z, Hiragana, Katakana, 24 Greek and 33 Cyrillic letters in both capital and small forms, Japanese Kanji, and 32 line drawing characters. There remain 1959 unallocated byte pairs that shall not be used.

- ISO-IR 168 (ESC 02/06 04/00 ESC 02/04 xx 04/02) : Japanese standard JIS X 0208:1990.

  This is a revision of ISO-IR 87 and is designated by the same escape sequences, preceded by the escape sequence that identifies a first revision. The revision introduces two additional characters. More information about the identification of revised registrations is given under escape sequences with intermediate bytes in the section on control functions.

- ISO-IR 159 (ESC 02/04 gg 04/04) : Japanese standard JIS X 0212:1990.

  This 94-position two-byte set contains 6067 characters that supplement those of ISO-IR 87 or ISO-IR 168. It provides 21 additional symbols, 27 additional Latin letters such as ø and þ, 171 Latin letters with diacritical marks, 21 Greek letters (final sigma and 20 letters with diacritical marks), 26 additional Cyrillic letters and 5801 additional Japanese Kanji characters.

- ISO-IR 149 (ESC 02/04 gg 04/03) : Korean standard KS C 5601:1987.

  This 94-position two-byte set contains 8224 characters that include 276 symbols, digits in both Arabic (0,1,...) and Roman (i,ii,... and I,II,...) forms, the Korean Hangul alphabet, Latin letters A-Z and a-z together with 11 additional capital letters and 16 additional small letters, 24 Greek and 33 Cyrillic letters in both capital and small forms, 68 line drawing characters, Japanese Hiragana and Katakana, 2350 Korean Hangul characters, 4888 Korean Hanja characters, and miscellaneous other characters such as vulgar fractions, superscripts and subscripts.

- ISO-IR 171 (ESC 02/04 gg 04/07) : Chinese standard CNS 11643:1986, Set 1.

  This 94-position two-byte set contains 6085 characters that include 234 symbols, digits in Arabic (0,1,...), Roman (i,ii,... and I,II,...) and Chinese forms, Latin letters A-Z and a-z, 24

Greek letters in both capital and small forms, 42 Mandarin phonetic symbols, 213 Chinese character radicals, 33 control code symbols such as “ESC” and “DEL” each as a single graphic, and 5401 of the most frequently used Chinese characters.

- ISO-IR 172 (ESC 02/04 gg 04/08) : Chinese standard CNS 11643:1986, Set 2.

  This 94-position two-byte set contains 7650 of the less frequently used Chinese characters.

In these escape sequences, replacement of “gg” by 02/08, 02/09, 02/10 or 02/11 specifies designation as a G0, G1, G2 or G3 code element respectively. Where “xx” has been used in place of “gg”, it denotes an exception to the current coding rules of ISO/IEC 2022 in that this bit combination is absent in the designation as a G0 code element. It is still replaced by 02/09, 02/10 or 02/11 to specify designation as a G1, G2 or G3 code element.

The Intermediate Bytes in these escape sequences identify designation of a 94-position two-byte character set as the code element concerned; see 5.3.4.4.

5.2.2.4 *Variable-length coding*

The existence of multiple-byte character sets leads to the possibility of variable-length coding. This may occur for two different reasons:

- the GL and GR code areas may contain character sets with differing coding lengths;
- if a 94 position set is invoked into the GL area then the characters SPACE and DELETE remain represented by the single bytes 02/00 and 07/15 even if the set invoked is a multiple-byte set.

When a character set is designated dynamically as the G0, G1, G2 or G3 element of a code by means of an escape sequence, the general syntax of such sequences allows the receiver to identify:

- whether the set is a 94 or 96 position set;
- whether it is a single-byte or multiple-byte set;
- if it is multiple-byte then how many bytes are required for each character.

This is described in more detail in section 5.3.

5.2.3 Combining characters

Another means of extending the repertoire of a character set beyond the number of available code positions is by means of combining characters. The original use of combining characters was to specify that certain characters of a code were to be non-spacing. When implemented on a receiving device such as a teleprinter, this had the effect of superposing the following character (a letter, say) on top of the non-spacing character (such as an accent) to produce a new character (in this example, an accented letter). A single non-spacing accent could therefore increase the repertoire of a code by many accented letters.

Although a non-spacing accent is classified as a graphic character in its own right, its coded representation cannot be used on its own to represent the accent concerned. It has to be followed by a SPACE character; superposition of the non-spacing accent on a non-printing space results in a normal (spacing) accent. This rule is stated explicitly in ISO/IEC 6937, which is the most well-known standard that uses non-spacing characters.

A non-spacing character is a combining character that combines with the following character. Now that the need to implement combining characters within electromechanical devices such as teleprinters

has receded, it has become possible also to specify (and implement) characters that combine with the preceding character. It is perhaps more natural, for example, to describe "é" as a small letter E with an acute accent above it than as an acute accent with a small letter E below it. This approach has been adopted for the new multiple-octet coded character set of ISO/IEC 10646. It is permitted also in the 7-bit and 8-bit code structure of ISO/IEC 2022 but has not, in fact, been used.

The use of combining characters brings variable-length coding into use even within a single code element.

5.3 Control Functions

The concept of a control function is an extension of that of a control character, such as CARRIAGE RETURN (CR), LINE FEED (LF), SHIFT IN (SI) and SHIFT OUT (SO), that has been present since the earliest development of character sets. A control character is simply a control function that is coded as a single bit combination. It is conventional for control functions to have both a name and an identifying acronym.

The code structure of ISO/IEC 2022, as described in section 4.2, includes two code elements C0 and C1 containing control functions. This section describes the standardized sources for these code elements and gives a brief account of the control functions that are available through their use. This account is aimed primarily at the use of control functions for code extension purposes.

The multiple-octet coded character set of ISO/IEC 10646 has a code structure that differs from that of ISO/IEC 2022. It includes its own specification of a code structure for graphic characters, but control functions are incorporated by a provision for the use of control functions encoded according to ISO/IEC 2022. Much of this section is therefore equally relevant to both ISO/IEC 2022 and ISO/IEC 10646 code structures.

5.3.1 Primary sets of control functions

The C0 code element of a code is known as its primary set of control functions. One specific C0 set is specified in ISO/IEC 6429. A code is not required by ISO/IEC 2022 to use this as its primary set, but if a primary set includes any of the control functions from the C0 set of ISO/IEC 6429 then it is required to have the same coding as in that standard. Alternative C0 sets are specified in the ISO 2375 Register.

The C0 set of ISO/IEC 6429 has its historical origin in the control characters of the ASCII character set. For this reason, 10 of the control functions of that set are transmission control functions such as START OF HEADER (SOH) that are not relevant to modern communications protocols. The semantics of those functions are specified in ISO/IEC 6429 by reference to a very old standard, ISO 1745, last revised in 1975.

The control functions of the C0 set of ISO/IEC 6429 are each represented by a single control character, i.e. they are coded by a single bit combination. With one exception the actions of these control functions are fully determined by that single control character. The exception is the ESCAPE (ESC) character, which is a control function whose semantic description in ISO/IEC 6429 is as follows:

- ESC causes the meanings of a limited number of bit combinations following it in the data stream to be changed.

The ESCAPE character together with this following sequence of bit combinations is known as an "escape sequence". The use of escape sequences is reserved by ISO/IEC 2022 to be for code extension purposes; see 5.3.3 below for more details. All primary sets are required by ISO/IEC 2022 to have the ESCAPE character at position 01/11.

5.3.2 Supplementary sets of control functions

The C1 code element of a code is known as its supplementary set of control functions. One specific C1 set is specified in ISO/IEC 6429. A code is not required by ISO/IEC 2022 to use this as its supplementary set, but a supplementary set is not permitted to include the ESCAPE character or any of the 10 transmission control characters of ISO 1745 described above concerning primary sets. Alternative C1 sets are specified in the ISO 2375 Register.

If the C1 code element is invoked into the CR area of an 8-bit code then its control functions are represented by a single control character, as for the C0 code element. Otherwise, and always for a 7-bit code, the control functions of the C1 code element are represented by an escape sequence.

The C1 set of ISO/IEC 6429 includes its own means of extension, similar to that provided by the ESCAPE character in the C0 set. The control function CONTROL SEQUENCE INTRODUCER (CSI) is followed by one or more bit combinations that together constitute a "control sequence". The permitted control sequences, and the functions they represent, are specified in ISO/IEC 6429 itself. This contrasts with escape sequences, whose use is specified by ISO/IEC 2022. Control sequences are primarily used for the control of devices for the display and presentation of character data.

The C1 set of ISO/IEC 6429 also includes provision for control strings, which are distinguished from escape and control sequences by having both an opening and a closing delimiter. The semantics of control strings is not standardized. They are used only where there is prior agreement between the sender and recipient of the data.

5.3.3 Escape sequences

5.3.3.1 *General construction*

The simplest coding of control functions by more than one bit combination is by means of an escape sequence. The general construction of an escape sequence is laid down in ISO/IEC 2022 and is as follows:

- An escape sequence consists of two or more bytes, of which the first is the ESCAPE character (coded 01/11) and the last, known as the Final Byte, is from columns 03 to 07 of the code table (but excluding the DELETE character 07/15).
- Any bytes between the ESCAPE and the Final Byte are known as Intermediate Bytes and are from column 02 of the code table.

This syntax ensures that an escape sequence can be delimited without any further knowledge of its syntax.

All standardized escape sequences are either defined in ISO/IEC 2022 or are specified in the International Register that is administered in accordance with ISO 2375. This International Register is the primary source of coded character sets for use as code elements in accordance with ISO/IEC 2022.

Escape sequences are further classified by the total number of bytes (bit combinations), including the ESCAPE character, that they involve.

5.3.3.2 *Two-byte escape sequences*

The two-byte escape sequences (those with no Intermediate Bytes) are classified into various types. The differing types are distinguished by the column of the code table that contains the Final Byte, as follows:

- Column 03 is reserved for private use where the control functions represented are agreed between the communicating parties.

- Columns 04 and 05 are reserved for representation of the functions of the C1 code element by escape sequences.

- Columns 06 and 07 are reserved for standardized control functions such as locking shifts that are registered, with their coded representations, in accordance with ISO 2375.

Always in a 7-bit code, and optionally in an 8-bit code, the control functions of a C1 code element are represented by two-byte escape sequences. The Final Byte is obtained by overlaying columns 04 and 05 of the code table with the C1 code element, as if the C1 code element were being temporarily invoked into these columns. For more detail of the use of the standardized control functions, see 5.3.4.

5.3.3.3 *Escape sequences with Intermediate Bytes*

Escape sequences with more than two bytes (those with Intermediate Bytes) are also classified into various types. The differing types are distinguished by the position within column 02 of the code table that contains the first Intermediate Byte. All these types are described below in more detail:

- Position 02/00 is reserved for Announcer Functions;

- Positions 02/01 and 02/02 are used for the designation of C0 and C1 sets of control functions;

- Position 02/03 is reserved for further control functions registered in accordance with ISO 2375 or for private use control functions;

- Position 02/04 is used for the designation of multiple byte graphic character sets;

- Position 02/05 is used to denote an escape from ISO/IEC 2022 code structure to a designated other coding system, such as the multiple-octet coded character set ISO/IEC 10646;

- Position 02/06 is used to identify revision of a registration under ISO 2375. It is followed by a final byte that identifies the revision number, with 04/00, 04/01, ... identifying the first, second, ... revision. Revision of a registration is only permitted under special circumstances laid down in ISO 2375, but see designation of C0 and C1 sets below for an example;

- Positions 02/07 and 02/12 are reserved for future standardization.

- Positions 02/08 to 02/11 and 02/13 to 02/15 are used for the designation of single-byte graphic character sets.

5.3.4 Code extension

5.3.4.1 *Locking shifts*

The primary means of invocation of the G0, G1, G2 and G3 code elements into the GL and GR areas of the code table is by means of locking shifts. Seven such locking shifts are required, since the G0 set cannot be invoked into the GR area. Two of these are included in the C0 set of ISO/IEC 6429 and are therefore required to have the same coding in every C0 set that includes them:

- LOCKING-SHIFT ZERO (LS0), which invokes the G0 set into the GL area, is coded as 00/15;

- LOCKING-SHIFT ONE (LS1), which invokes the G1 set into the GL area, is coded as 00/14.

For historical reasons, when used with a 7-bit code these are known instead as SHIFT-IN (SI) and SHIFT-OUT (SO) respectively.

The remaining five locking shifts are represented by standardized escape sequences. Together with their registration numbers in the ISO 2375 Register, they are:

- LOCKING-SHIFT TWO (LS2), which invokes the G2 set into the GL area, which is ISO-IR 62 and coded as ESC 06/14;
- LOCKING-SHIFT THREE (LS3), which invokes the G3 set into the GL area, which is ISO-IR 63 and coded as ESC 06/15;
- LOCKING-SHIFT ONE RIGHT (LS1R), which invokes the G1 set into the GR area, which is ISO-IR 66 and coded as ESC 07/14;
- LOCKING-SHIFT TWO RIGHT (LS2R), which invokes the G2 set into the GR area, which is ISO-IR 65 and coded as ESC 07/13;
- LOCKING-SHIFT THREE RIGHT (LS3R), which invokes the G3 set into the GR area, which is ISO-IR 64 and coded as ESC 07/12.

5.3.4.2 *Single shifts*

The C1 set of ISO/IEC 6429 includes non-locking shifts SINGLE-SHIFT TWO (SS2) and SINGLE-SHIFT THREE (SS3) that are used to invoke the G2 and G3 code elements for the next graphic character only. It is a matter for prior agreement as to whether these sets are invoked into the GL or GR areas by these single shifts. The area selected is known as the single-shift area. The announcer functions of ISO/IEC 2022 may be used to form this agreement.

It is permitted by ISO/IEC 2022 to include these non-locking shifts in a primary (C0) set of control functions. One C0 set that includes them is the set ISO-IR 106, the Teletex primary set of Control Functions of CCITT Recommendation T.61, which is contained in the ISO 2375 register.

5.3.4.3 *Designation of sets of control functions*

Besides the C0 and C1 sets of ISO/IEC 6429, other standardized sets of control functions are specified in the ISO 2375 register. Although this is nominally a register of standardized escape sequences, where these escape sequences are used to designate coded character sets as elements of a 7-bit or 8-bit code then the register includes the specification of that code element. Escape sequences commencing ESC 02/01 and ESC 02/02 designate specific sets of control functions as the C0 and C1 element respectively. As examples:

- The C0 set of ISO/IEC 6429 is registered as ISO-IR 1 and designated by ESC 02/01 04/00;
- The C1 set of ISO/IEC 6429 is registered as ISO-IR 77 and designated by ESC 02/02 04/03. On its own this designates the C1 set of the 1983 (second) edition of ISO/IEC 6429. To identify the C1 set of the 1992 (third) edition, this escape sequence must be immediately preceded by ESC 02/06 04/00 to identify the first revision.

5.3.4.4 *Designation of sets of graphic characters*

By far the largest part of the ISO 2375 register is the specification of sets of graphic characters that may be designated by means of escape sequences. More information on these sets is given in section 5.2. Individual sets are designated as the G0, G1, G2 or G3 code element by an escape sequence that describes, by means of Intermediate Bytes as specified in ISO/IEC 2022, the nature of the character set and the code element to which it is being invoked. The Final Byte identifies the actual character set concerned.

For single-byte character sets, one Intermediate Byte identifies the code element as follows:

- Bytes 02/08 to 02/11 designate a set of 94 positions (i.e. 02/01 to 07/14 if invoked in the GL area) as the G0, G1, G2 or G3 code element respectively.

- Bytes 02/13 to 02/15 designate a set of 96 positions (i.e. 02/00 to 07/15 if invoked in the GL area) as the G1, G2 or G3 code element respectively. Such a set is not permitted as a G0 code element.

For multiple-byte character sets, the first Intermediate Byte is 02/04 and the second Intermediate Byte identifies the code element as follows:

- Bytes 02/08 to 02/11 designate a multi-byte 94-set as the G0, G1, G2 or G3 code element respectively.

- Bytes 02/13 to 02/15 designate a a multibyte 96-set as the G1, G2 or G3 code element respectively. Such a set is not permitted as a G0 code element.

- Exceptionally, for three registrations only, byte 02/04 is followed directly by a Final Byte to designate a two-byte set of 94 positions as a G0 code element. The Final Bytes permitted for this exceptional case are 04/00, 04/01 and 04/02. The exception dates from an early edition of ISO/IEC 2022 in which such a set could be invoked as a G0 code element but not as a G1, G2 or G3 code element.

Further Intermediate Bytes may also be present in the escape sequence. They are used, for example, to identify the number of bytes per character in a multiple-byte character set. A receiving implementation is therefore able to parse a received data stream into characters without the need for detailed knowledge of the contents of the ISO 2375 Register.

5.3.4.5 *Announcement functions*

Provision is made in ISO/IEC 2022 for the announcement, by means of escape sequences, of a wide range of options permitted by that standard. All these escape sequences consist of ESC 02/00 followed by a Final Byte. Examples are:

- ESC 02/00 04/06 to announce that, in an 8-bit code, the C1 code element shall not be invoked into the CR area. Its control functions are therefore coded as escape sequences;

- ESC 02/00 04/07 to announce that, in an 8-bit code, the C1 code element shall be invoked into the CR area. Its control functions are therefore coded by single bytes;

- ESC 02/00 05/12 to announce that, in an 8-bit code, the single-shift area shall be the GR area.

5.3.5 Control sequences

Control sequences are defined in ISO/IEC 6429 and are used to represent many of the control functions that are specified in that standard. The general construction of a control sequence is similar to that of an escape sequence but it contains a refinement to permit the representation of control functions that require parameters:

- A control sequence consists of the control function CONTROL FUNCTION INTRODUCER (CSI) followed by one or more bytes, of which the last, known as the Final Byte, is from columns 04 to 07 of the code table (but excluding the DELETE character 07/15).

- The bytes, if any, between the CSI and the Final Byte consist of zero or more Parameter Bytes followed by zero or more Intermediate Bytes. Parameter Bytes are from column 03 of the code table and Intermediate Bytes are from column 02.

The CSI control function is present in the C1 set of ISO/IEC 6429 and is coded either by the single bit combination 09/11 or by the escape sequence ESC 05/11 (see 5.3.2 above). Note that the position of CSI in the C1 set corresponds precisely to the position of ESC in the C0 set.

The function represented by a control sequence is determined by the Final Byte together with any Intermediate Bytes that may be present. The Parameter Bytes act solely as parameters of the function so determined. The syntax of the Parameter Bytes is as follows:

- If the first Parameter Byte is in the range 03/00 to 03/11 then the entire sequence of Parameter Bytes is required to consist of bytes from this range. It represents the coding, according to the ISO-IR 6 (ASCII) character set, of a sequence composed of digits ZERO to NINE together with COLON and SEMICOLON. A SEMICOLON separates parameters when the control function takes more than one parameter. Each parameter is expressed as a decimal number in which a COLON is used to separate the integer and fractional parts.

- If the first Parameter Byte is in the range 03/12 to 03/15 then there is no standardized interpretation for the sequence of Parameter Bytes.

An example of a control function that takes a single numeric parameter is:

- IDENTIFY GRAPHIC SUBREPERTOIRE (IGS), coded as CSI nn 02/00 04/13.

This control function identifies a subrepertoire of the graphic characters of ISO/IEC 10367 which is registered in accordance with ISO/IEC 7350. In the coded representation, “nn” represents the registration number of the repertoire in the ISO/IEC 7350 Register.

5.3.6 Control strings

The C1 set of ISO/IEC 6429 includes provision for control strings that have no standardized meaning but which can be used by private agreement for various control purposes. Each control string has an opening delimiter, contained in the C1 set, that indicates the general nature of the control purpose. The available opening delimiters are:

- Application Program Command (APC)
- Device Control String (DCS)
- Operating System Command (OSC)
- Privacy Message (PM)
- Start of String

All control strings are terminated by a common closing delimiter from the C1 set, namely:

- String Terminator (ST)

Between the two delimiters there may be any sequence of bit combinations other than those representing the delimiters SOS and ST.

5.3.7 Control functions for text communication

The control functions specified in ISO/IEC 6429 contain many functions primarily intended for the control of devices for the display and presentation of character data. These can be used for communicating page layout, either in a fixed format or in a form to allow automatic reformatting when the sender and receiver use different fonts. However, the specifications of the control functions

need refinement to allow this to be achieved most satisfactorily. A specification of control functions from ISO/IEC 6429 customised for use in page image communication is given in ISO/IEC 10538.

6 Guides to standards

All the standards described in this annex have a separate description within this section.

6.1 International Standards

6.1.1 ISO/IEC 646

6.1.1.1 *Current edition*

ISO/IEC 646:1991, Information technology – ISO 7-bit coded character set for information interchange (third edition).

6.1.1.2 *Description*

This standard contains the specification of a G0 set of 94 graphic characters for use in the GL area of a 7-bit code. Use of the code requires in addition a C0 set of control functions to be invoked in the CL area. The standard requires this set of control functions to be a subset of the C0 set of ISO/IEC 6429.

The specification contains a number of options. Of the 94 code positions for graphic characters, 10 have no specific character allocated to them. These positions are available for national or application-oriented use. A further two positions have two alternative allocations.

6.1.1.3 *Tutorial guidance*

This standard introduces the concept of a *version* of ISO/IEC 646. A version is obtained by:

- specifying the C0 set of control functions;
- allocating a specific graphic character to each of the 10 unallocated code positions, or declaring that position to be unused;
- making a specific choice between the alternatives NUMBER SIGN and POUND SIGN for position 02/03 and between DOLLAR SIGN and CURRENCY SIGN for position 02/04.

The standard specifies one version itself. This is known as the International Reference Version (IRV) of ISO/IEC 646. Its C0 set is the C0 set of ISO/IEC 6429 and its G0 set is that registered as ISO-IR 6 in the ISO 2375 Register. This is the set commonly known as ASCII. In positions 02/03 and 02/04 it specifies the NUMBER SIGN and DOLLAR SIGN respectively. It is important to note that this is a change from the IRV of the second edition ISO 646:1983, which specified the CURRENCY SIGN in position 02/04 and was registered as ISO-IR 2.

The G0 sets of a number of other versions of ISO/IEC 646 are also registered in accordance with ISO 2375. The register entries of a selection of these, together with the escape sequences that designate them as a G0, G1, G2 or G3 code element, are as follows:

- ISO-IR 2 (ESC gg 04/00) : International Reference Version of ISO 646:1983
- ISO-IR 4 (ESC gg 04/01) : British version
- ISO-IR 6 (ESC gg 04/02) : International Reference Version of ISO/IEC 646:1991 (also USA version)

- ISO-IR 10 (ESC gg 04/07) : Swedish version
- ISO-IR 11 (ESC gg 04/08) : Swedish version for names
- ISO-IR 14 (ESC gg 04/10) : Japanese version for Roman Characters
- ISO-IR 15 (ESC gg 05/09) : Italian version
- ISO-IR 16 (ESC gg 04/12) : Portuguese version, ECMA (Olivetti)
- ISO-IR 17 (ESC gg 05/10) : Spanish version, ECMA (Olivetti)
- ISO-IR 21 (ESC gg 04/11) : German version
- ISO-IR 60 (ESC gg 06/00) : Norwegian version
- ISO-IR 69 (ESC gg 06/06) : French version
- ISO-IR 84 (ESC gg 06/07) : Portuguese version, ECMA (IBM)
- ISO-IR 85 (ESC gg 06/08) : Spanish version, ECMA (IBM)
- ISO-IR 86 (ESC gg 06/09) : Hungarian version
- ISO-IR 141 (ESC gg 07/10) : Serbocroatian and Slovenian version
- ISO-IR 170 (ESC gg 02/01 04/02) : Invariant characters of ISO/IEC 646 (82 characters)

In these escape sequences, replacement of "gg" by 02/08, 02/09, 02/10 or 02/11 specifies designation as a G0, G1, G2 or G3 code element respectively. These Intermediate Bytes specify designation of a 94-position single-byte character set as the code element concerned; see designation of graphic character sets in the section on control functions.

For reasons of backward compatibility with previous versions, ISO/IEC 646 permits the use of BACKSPACE (BS) and CARRIAGE RETURN (CR) control functions to create composite characters. In particular, it specifies that the sequence of a letter character, BACKSPACE and one of QUOTATION MARK, APOSTROPHE or COMMA should be interpreted as that letter bearing a diaeresis, acute accent or cedilla respectively. The character set separately includes GRAVE ACCENT, CIRCUMFLEX ACCENT and TILDE which may be combined similarly to produce letters with other diacritical marks. More recent character set standards that permit characters to be combined, such as ISO/IEC 6937, make use of specific combining characters as described in the section on graphic characters, so avoiding the use of control functions.

6.1.2 ISO/IEC 2022

6.1.2.1 *Current edition*

ISO/IEC 2022:1994, Information technology – Character code structure and extension techniques (fourth edition).

6.1.2.2 *Description*

This standard specifies a structure for 7-bit and 8-bit codes that is adopted by all such codes produced under the auspices of ISO/IEC JTC1/SC2. This is the subcommittee entrusted jointly by ISO and IEC with the development of character set coding matters.

This standard also specifies means by which the correspondence between bit combinations and characters may be changed during a particular instance of information interchange. This is known as

code extension. It makes use of control functions that are themselves represented by bit combinations within the original code.

6.1.2.3 *Tutorial guidance*

This annex contains introductions to the features of ISO/IEC 2022 at various levels:

- The basic ideas of ISO/IEC 2022 are described in section 4.2, Concepts and terminology.
- Rurther details can be found in section 5.2, Graphic character sets, and 5.3, Control Functions.

The facilities of ISO/IEC 2022 are of a powerful and general nature. A simplified structure for 8-bit codes and code extension is specified in ISO/IEC 4873. The structure of ISO/IEC 4873 does not permit multiple-byte coded character sets to be invoked and therefore excludes, for example, the coding of Japanese, Chinese and Korean ideographic characters as described in 5.2.

6.1.3 ISO 2375

6.1.3.1 *Current edition*

ISO 2375:1985, Data processing – Procedure for registration of escape sequences (third edition).

6.1.3.2 *Description*

The structure of 7-bit and 8-bit codes for the representation of character sets, and code extension techniques for use with such character sets, are specified in ISO/IEC 2022. These code extension techniques make use of escape sequences, a concept defined in ISO/IEC 2022. That standard defines classes of escape sequences but does not assign particular meanings to individual escape sequences. ISO 2375 specifies the procedures to be followed in preparing and maintaining a register of specific escape sequence meanings.

6.1.3.3 *Tutorial guidance*

The procedures of ISO 2375 allow for the registration of an escape sequence, for the withdrawal of a registration by the authority that sponsored it, and in exceptional circumstances for the revision of a registration.

The registration authority for ISO 2375 is:

- IPSJ/ITSCJ (Information Processing Society of Japan/Information Technology Standards Commission of Japan) Room 308-3, Kikai-Shinko-Kaikan Bldg., 3-5-8, Shiba-Koen, Minato-ku, Tokyo 105 JAPAN

The register used to be available free of charge in paper form. However, the register is now available in a much more convenient form electronically on the Web. See 6.2.1 on the ISO 2375 Register for more details of its contents.

More detail of the general classification of escape sequences is given in section 5.3.

6.1.4 ISO/IEC 4873

6.1.4.1 *Current edition*

ISO/IEC 4873:1991, Information technology – ISO 8-bit code for information interchange – Structure and rules for implementation (third edition).

6.1.4.2 *Description*

This standard specifies a structure for 8-bit codes that builds on the general structure for such codes laid down in ISO/IEC 2022. In particular the content of the GL area of the code table is fully specified and the content of the GR area is restricted to be a character set that makes use of single-byte coding (and so contains at most 96 characters). The fixed content for the GL area is the set registered in the ISO 2375 Register as ISO-IR 6. This set is also the International Reference Version (IRV) of ISO/IEC 646:1991 and is more commonly known as the ASCII character set.

The code extension techniques permitted by ISO/IEC 4873 are only a selection of those specified by ISO/IEC 2022.

6.1.4.3 *Tutorial guidance*

ISO/IEC 4873 specifies three levels of implementation:

- Level 1 requires a C0 set in addition to the fixed (ASCII) G0 set and it permits a C1 set and a G1 set to be present. It does not permit the use of a G2 or G3 set. The C1 and G1 sets, if present, are invoked in the CR and GR areas.

- Level 2 includes the facilities of level 1 but permits, in addition, a G2 and G3 set that may be invoked by means of the single-shift control functions SS2 and SS3. The C1 set is required to be present and to contain at least the SS2 and SS3 control functions at positions 08/14 and 08/15 respectively. A C1 set containing only these two control functions is registered as ISO-IR 105. The use of locking shifts is not permitted.

- Level 3 includes the facilities of level 2 but permits, in addition, the use of the locking shifts LS1R, LS2R and LS3R.

The coding of single shifts and locking shifts is given in section 5.3.

A collection of coded graphic character sets suitable for use within the structure of ISO/IEC 4873 has been standardized in ISO/IEC 10367.

6.1.5 ISO/IEC 6429

6.1.5.1 *Current edition*

ISO/IEC 6429:1992, Information technology – Control functions for coded character sets (third edition).

6.1.5.2 *Description*

This standard specifies a repertoire of a large number of control functions, giving both their definitions and their coded representations. It includes a C0 set and a C1 set that may be designated for use with any 7-bit or 8-bit code that conforms to the code structure laid down in ISO/IEC 2022, or with the universal multiple-octet coded character set of ISO/IEC 10646. The coded representation of individual control functions consists either of:

- a single bit combination from either the C0 or the C1 set, or

- a sequence of bit combinations that starts with one from either the C0 or C1 set and which is constructed according to specific rules that allow the end of the sequence to be determined without detailed knowledge of the syntax of the particular function represented.

6.1.5.3 *Tutorial guidance*

The control functions defined in ISO/IEC 6429 fall into a number of distinct categories. They are primarily concerned with:

- code extension;
- transmission control for coded data;
- formatting, editing and presentation of character data;
- control of ancillary devices.

More detail of the coding methods used, and of the control functions available for code extension, is given in 5.3 on control functions.

The control functions for formatting, editing and presentation of character data are suitable for use in communicating page layout, either in a fixed format or in a form to allow automatic reformatting when the sender and receiver use different fonts. More specific definitions of these functions, tailored to these particular uses, are given in ISO/IEC 10538; see 6.1.10.

6.1.6 ISO/IEC 6937

6.1.6.1 *Current edition*

ISO/IEC 6937:1994, Information technology – Coded graphic character set for text communication – Latin alphabet (second edition).

6.1.6.2 *Description*

This standard contains the specification of a set of graphic characters for the GL and GR areas of an 8-bit code table. Provision is also included for it to be used as a 7-bit code by making use of code extension techniques from ISO/IEC 2022.

The GL area contains the G0 set of ISO/IEC 4873, namely ISO-IR 6, the ASCII character set. The GR area contains characters in 86 of the 96 available code positions, the remaining 10 positions being excluded from use. The 13 characters in column 12, there being three unassigned positions in this column, are non-spacing diacritical marks (combining characters). The characters in all other columns are spacing (non-combining) characters.

Due to its use of non-spacing diacritical marks, the code can represent more characters than there are code positions. The standard includes a normative specification of the 333 graphic characters, including the SPACE character, that it is permitted to represent by use of the 181 bit combinations that are assigned in the code. This set of 333 graphic characters constitutes the repertoire of the standard.

6.1.6.3 *Tutorial guidance*

ISO/IEC 6937 specifies a character set that is primarily intended for information interchange using the Latin script. Characters with diacritical marks (accents) are transmitted by sending a non-spacing accent character (the combining character) followed by the underlying letter character. The available diacritical marks are:

- grave, acute and circumflex accents;
- tilde, macron, breve, diaeresis, cedilla, ogonek and caron;
- ring above and dot above;

- double acute accent.

The approach adopted by this standard originates with electromechanical devices such as teleprinters. In such devices it is not difficult to prevent certain characters from operating the escapement mechanism that moves to the next printing position. It is an approach that is unsuitable for most data and text processing applications. It does have advantages, however, for sorting (collating) purposes since the underlying letter is easily identified.

The repertoire is suitable for use with a wide range of languages. In particular it contains all the characters of all of the codes known as Latin Alphabets Nos. 1 to 6 that are defined in parts of ISO/IEC 8859, and some additional characters as well. However, it does not contain some of the characters of the codes known as Latin Alphabets Nos. 7 to 9 that are defined in other parts of ISO/IEC 8859. There are no plans to add these missing characters, in particular, the EURO SIGN. It states that it covers the following languages:

- Afrikaans, Albanian, Basque, Breton, Catalan, Croat, Czech, Danish, Dutch. English, Esperanto, Estonian, Faroese, Finnish, French, Frisian, Galician, German, Greenlandic, Hungarian, Icelandic, Irish, Italian, Lapp (Sami), Latvian, Lithuanian, Maltese, Norwegian, Occitan, Polish, Portuguese, Rhaeto-Romanic, Roumanian, Scots Gaelic, Slovak, Slovene, Sorbian, Spanish, Swedish, Turkish and Welsh

However, it also provides an informative note that it does not cover the full repertoire required for Welsh. The missing characters are W and w with acute and grave accents and with diaeresis, and Y and y with grave accents. These characters are all included in ISO-IR 182, which may be used with ISO-IR 6 to form a Welsh variant of Latin Alphabet No.1. They are also included in the newer Latin Alphabet No. 8 defined in ISO/IEC 8859-14 together with missing characters needed for other Celtic languages. See 6.1.8, the guide to ISO/IEC 8859, for more details.

6.1.7 ISO/IEC 7350

6.1.7.1 *Current edition*

ISO/IEC 7350:1991, Information technology – Registration of repertoires of graphic characters from ISO/IEC 10367 (second edition).

6.1.7.2 *Description*

ISO/IEC 7350 specifies the procedures to be followed in preparing, publishing and maintaining a register of graphic character repertoires which are composed entirely of graphic characters from ISO/IEC 10367. The coded representation of the characters of such repertoires is not prescribed by the entries in the register.

6.1.7.3 *Tutorial guidance*

A repertoire registered in accordance with ISO/IEC 7350 is required to consist of either:

- the characters of the G0 set of ISO/IEC 10367, and
- the characters of any number of its supplementary sets except that of ISO/IEC 6937

or:

- the characters of the G0 set of ISO/IEC 10367, and
- the characters of any number of the supplementary sets chosen from the Arabic, Cyrillic, Greek, Hebrew or Basic Box Drawing supplementary sets, and

- a subrepertoire of the repertoire associated with the supplementary set of ISO/IEC 6937.

ISO/IEC 7350 requires a numeric identifier to be assigned to each registered repertoire. This identifier is intended for use with the control function IDENTIFY GRAPHIC SUBREPERTOIRE (IGS) that is defined in ISO/IEC 6429. The coding of this control function and its integer parameter by a control sequence is described in section 5.3.

6.1.8 ISO/IEC 8859

6.1.8.1 *Current edition*

This is a multi-part standard.

ISO 8859, Information processing – 8-bit single-byte coded graphic character sets

— Part 1: Latin alphabet No.1 (1997, second edition)
— Part 2: Latin alphabet No.2 (1998, second edition)
— Part 3: Latin alphabet No.3 (1998, second edition)
— Part 4: Latin alphabet No.4 (1998, second edition)
— Part 5: Latin/Cyrillic alphabet (1998, second edition)
— Part 6: Latin/Arabic alphabet (1998, second edition)
— Part 7: Latin/Greek alphabet (1998, second edition)
— Part 8: Latin/Hebrew alphabet (1998, second edition)
— Part 9: Latin alphabet No.5 (1998, second edition)
— Part 10: Latin alphabet No.6 (1998, second edition)
— Part 11: Latin/Thai alphabet (1998)
— Part 12: Unassigned
— Part 13: Latin alphabet No.7 (1998)
— Part 14: Latin alphabet No.8 (1998)
— Part 15: Latin alphabet No.9 (1998)

6.1.8.2 *Description*

Each part of this standard contains the specification of a set of graphic characters for the GL and GR areas of an 8-bit code table. For each part, the GL area contains the G0 set of ISO/IEC 4873 (ISO-IR 6, the ASCII character set), so that only the 96 character positions of the GR area vary between the parts.

The GR area makes use of single-byte coding and contains no non-spacing diacritical marks (or other combining characters). The repertoire of the code therefore comprises at most 191 graphic characters including the SPACE character, one for each of the 191 available code positions.

6.1.8.3 *Tutorial guidance*

Each part of ISO/IEC 8859 specifies a character set that is suitable both for data and text processing applications and for information interchange.

The GR area of each of the Latin Alphabets includes a selection of accented Latin letters, and possibly also additional Latin letters such as Icelandic letters Þ (capital letter thorn), þ (small letter thorn) and ð (small letter eth). Each of these parts is suitable for multiple-language applications using the Latin script. The remaining five parts contain characters from a non-latin script in the GR area, as indicated by their title.

Each part specifying a Latin Alphabet lists the languages for which it has been designed. These are:

Latin Alphabet No.1

Albanian, Basque, Breton, Catalan, Danish, Dutch, English, Faroese, Finnish, French (with restrictions), Frisian, Galician, German, Greenlandic, Icelandic, Irish Gaelic (new orthography), Italian, Latin, Luxemburgish, Norwegian, Portuguese, Rhjaeto-Romanic, Scotish Gaelic, Spanish and Swedish.

Latin Alphabet No.2

Albanian, Croat, Czech, English, German, Hungarian, Latin, Polish, Romanian, Slovak, Slovene and Sorbian.

Latin Alphabet No.3

Esperanto and Maltese, and if needed in conjunction with these, English, French (with restrictions), German, Italian, Latin and Portuguese. Coding of Turkish characters is deprecated in this code.

Latin Alphabet No.4

Danish, English, Estonian, Finnish, German, Greenlandic, Latin, Latvian, Lithuanian, Norwegian, Sámi (with restrictions), Slovene and Swedish.

Latin Alphabet No.5

Albanian, Basque, Breton, Catalan, Danish, Dutch, English, Faroese, Finnish, French (with restrictions), Frisian, Galician, German, Greenlandic, Irish Gaelic (new orthography), Italian, Latin, Luxemburgish, Norwegian, Portuguese, Spanish, Rhaeto-Romanic, Scottish Gaelic, Spanish, Swedish and Turkish.

Latin Alphabet No.6

Danish, English, Estonian, Faroese, Finnish, German, Greenlandic, Icelandic, Irish Gaelic (new orthography), Latin, Lithuanian, Norwegian, Sámi (with restrictions), Slovene and Swedish.

Latin Alphabet No.7

Danish, English, Estonian, Finnish, German, Latin, Latvian, Lithuanian, Norwegian, Polish, Slovene and Swedish.

Latin Alphabet No.8

Albanian, Basque, Breton, Catalan, Cornish, Danish, Dutch, English, French (with restrictions), Frisian, Galician, German, Greenlandic, Irish Gaelic (old and new orthographies), Ialian, Latin, Luxemburgish, Manx Gaelic, Norwegian, Portuguese, Rhaeto-Romanic, Scottish Gaelic, Spanish, Swedish and Welsh.

Latin Alphabet No.9

Albanian, Basque, Breton, Catalan, Danish, Dutch, English, Estonian, Faroese, Finnish, French, Frisian, Galician, German, Greenlandic, Icelandic, Irish Gaelic (new orthography), Italian, Latin, Luxemburgish, Norwegian, Portuguese, Rhaeto-Romanic, Scotish Gaelic, Spanish and Swedish

For writing French, three characters not included in Latin Alphabets 1, 3, 5 and 8, are also needed. These are included in Latin Alphabet No.9.

The Skolt Sámi dialect, and older Sámi orthography, require certain additional characters. These have been registered in ISO-IR 158 and ISO-IR 197 in the ISO 2375 Register. ISO/IEC 8859-10

recommends the use of that character set as a G2 or G3 code element together with the GL and GR sets of ISO/IEC 8859-10 as G0 and G1 code elements when these characters are required.

ISO-IR 182 was registered to be an alternative GR set for ISO/IEC 8859-1 to cover the Welsh language. Use of Latin Alphabet No.8 is now recommended for that purpose.

The coded character sets of the GR areas of each part of ISO/IEC 8859 are included in the ISO 2375 Register. With the exception of parts 10 to 15 they are also included in ISO/IEC 10367. For the registration number and escape sequences assigned to these sets, see the guide to ISO/IEC 10367 in 6.1.9.

6.1.9 ISO/IEC 10367

6.1.9.1 *Current edition*

ISO/IEC 10367:1991, Information technology – Standardized coded graphic character sets for use in 8-bit codes (first edition).

6.1.9.2 *Description*

ISO/IEC 10367 specifies a collection of coded graphic character sets suitable for use within the structure of an 8-bit code as laid down in ISO/IEC 4873. These sets are all suitable for use as any of the code elements G1, G2 and G3 in a version of ISO/IEC 4873 at any of its three levels of implementation. The G0 code element of ISO/IEC 4873 is prescribed by that standard but is repeated for information in ISO/IEC 10367. ISO/IEC 10367 does not specify the sets C0 and C1 of control functions that may be used in a version of ISO/IEC 4873 that conforms to ISO/IEC 10367.

6.1.9.3 *Tutorial guidance*

The coded graphic character sets specified in ISO/IEC 10367 are all contained in the ISO 2375 Register. Their register entries, together with the escape sequences that designate them as a G1, G2 or G3 code element, are as follows.

- For the G0 code element:
    - ISO-IR 6 (ESC 02/08 04/02) : International Reference Version of ISO/IEC 646:1991
- For the G1, G2 and G3 code elements:
    - ISO-IR 100 (ESC gg 04/01) : Latin Alphabet No.1, Supplementary Set (GR area of ISO 8859-1)
    - ISO-IR 101 (ESC gg 04/02) : Latin Alphabet No.2, Supplementary Set (GR area of ISO 8859-2)
    - ISO-IR 109 (ESC gg 04/03) : Latin Alphabet No.3, Supplementary Set (GR area of ISO 8859-3)
    - ISO-IR 110 (ESC gg 04/04) : Latin Alphabet No.4, Supplementary Set (GR area of ISO 8859-4)
    - ISO-IR 148 (ESC gg 04/13) : Latin Alphabet No.5, Supplementary Set (GR area of ISO/IEC 8859-9)
    - ISO-IR 144 (ESC gg 04/12) : Cyrillic Supplementary Set (GR area of ISO/IEC 8859-5)
    - ISO-IR 127 (ESC gg 04/07) : Arabic Supplementary Set (GR area of ISO 8859- 6)

- ISO-IR 126 (ESC gg 04/06) : Greek Supplementary Set (GR area of ISO 8859- 7)
- ISO-IR 138 (ESC gg 04/08) : Hebrew Supplementary Set (GR area of ISO 8859- 8)
- ISO-IR 154 (ESC gg 05/00) : Supplementary Set for Latin Alphabets No.1 or No.5, and No.2
- ISO-IR 155 (ESC gg 05/01) : Basic Box Drawing Set
- ISO-IR 156 (ESC gg 05/02) : Supplementary Set of ISO/IEC 6937

Since the publication of ISO/IEC 10367, other character sets have been registered that are also intended for use as G1, G2 or G3 code elements in a version of ISO/IEC 4873. Some of these have also been standardised in further parts of ISO/IEC 8859. Although these are not part of ISO/IEC 10367, they are listed here for completeness:

- ISO-IR 157 (ESC gg 05/06) : Latin Alphabet No.6, Supplementary Set (GR area of ISO/IEC 8859-10)
- ISO-IR 166 (ESC gg 05/04) : Thai Supplementary Set (GR area of ISO/IEC 8859-11)
- ISO-IR 179 (ESC gg 05/09) : Latin Alphabet No.7, Baltic Rim Supplementary Set (GR area of ISO/IEC 8859-13)
- ISO/IR 199 (ESC gg 05/15) : Latin Alphabet No.8, Celtic Suplementary Set (GR area of ISO/IEC 8859-14)
- ISO-IR 203 (ESC gg 06/02) : Latin Alphabet No.9, European Supplementary Set (GR Area of ISO/IEC 8859-15)
- ISO-IR 158 (ESC gg 05/08) : Supplementary Set for Sami (Lappish) to complement Latin Alphabet No.6 (from annex A of ISO/IEC 8859-10)
- ISO-IR 197 (ESC gg 05/13) : Supplementary Set for Sami to complement Latin Alphabet No.6 (from annex A of ISO/IEC 8859-10)
- ISO-IR 182 (ESC gg 05/12) : Welsh variant of Latin Alphabet No.1, Supplementary Set

In these escape sequences, replacement of “gg” by 02/13, 02/14 or 02/15 specifies designation as a G1, G2 or G3 code element respectively. These Intermediate Bytes specify designation of a 96-position single-byte character set as the code element concerned; see 5.3.4.4. on the designation of graphic character sets.

For more details of the character sets taken from ISO/IEC 646, ISO/IEC 8859 and ISO/IEC 6937, including the languages for which they are suitable, see the entries for those standards in this annex. The entry for ISO/IEC 8859 also covers the additional sets listed above.

There is a requirement concerning the Supplementary Set of ISO/IEC 6937 (ISO- IR 156) that it shall not be used in conjunction with any of the Latin Alphabet Supplementary Sets. However, it may be used in conjunction with any two of the supplementary sets for Greek, Cyrillic, Arabic and Hebrew as G2 and G3 code elements, to provide support for these scripts in addition to a wide range of languages in the Latin script.

Use of ISO-IR 156 requires the support of non-spacing diacritical marks and so results in a code with variable-length coding. All the 333 characters (including SPACE) that are in the repertoire of

ISO/IEC 6937 can be represented with single-byte coding by selecting ISO-IR 100 or 148 as the G1 code element, ISO-IR 101 as the G2 code element and ISO-IR 154 as the G3 code element.

There may be a need in an instance of communication to be able to identify a subrepertoire of the full repertoire of characters present in the character sets of ISO/IEC 10367. Procedures for the registration of such subrepertoires are specified in ISO/IEC 7350.

6.1.10 ISO/IEC 10538

6.1.10.1 *Current edition*

ISO/IEC 10538:1991, Information technology – Control functions for text communication (first edition).

6.1.10.2 *Description*

ISO/IEC 10538 defines the control functions, and their coded representations, needed for use in text communication. The coded representations are intended for use when the control functions concerned are embedded in the communicated text, not when they are separated from the text as elements of a communication protocol.

6.1.10.3 *Tutorial guidance*

ISO/IEC 10538 is divided into three sections. The first section provides a general introduction. The second section specifies control functions for text in a page-image format in which the sender's and recipient's pages are intended to be identical. The third section specifies control functions for text that may be either in a formatted or a reformattable form, suitable for use where the sender's and recipient's fonts differ.

With two exceptions, the control functions of ISO/IEC 10538 have been taken from ISO/IEC 6429 but they have been given more specific definitions than in that standard. The exceptions are the functions PAGE TERMINATOR (PT) and DOCUMENT TERMINATOR (DT). These are alternative names assigned by ISO/IEC 10538 to the control functions INFORMATION SEPARATOR THREE (IS3) and INFORMATION SEPARATOR FOUR (IS4) of ISO/IEC 6429, to represent their correspondingly more specific definitions.

6.1.11 ISO/IEC 10646

6.1.11.1 *Current edition*

ISO/IEC 10646-1:1993, Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane (first edition).

6.1.11.2 *Description*

ISO/IEC 10646 specifies a multiple-octet coded character set, the UCS, that is applicable to the representation, transmission, interchange, processing, storage, input and presentation of the written form of the languages of the world as well as additional symbols. It uses a coding system different from that specified in ISO/IEC 2022, but it provides mechanisms in accordance with ISO/IEC 2022:

- to designate the use of the UCS from ISO/IEC 2022, and
- to designate return to ISO/IEC 2022 from the UCS.

6.1.11.3 *Tutorial guidance*

ISO/IEC 10646 is planned as a standard in multiple parts, of which only part 1 has so far been published. In its full form a four-octet coding (32 bits) will be required, but a two-octet (16 bit) coding is specified that covers use of the Basic Multilingual Plane given in Part 1.

Annex B contains a detailed description to the architecture and content of ISO/IEC 10646 (UCS).

6.1.12 ISO/IEC ISP 12070

6.1.12.1 *Current edition*

ISO/IEC ISP 12070-1:1996, Information technology – International Standardized Profiles FCSnnn – Character Set – Code Structure based on ISO/IEC 2022 – Part 1: FCS111 – 2022 Option 1 (first edition).

6.1.12.2 *Description*

The profile ISO/IEC ISP 12070 forms part of the documentation of Open Systems Interconnection (OSI), for which the Basic Reference Model is specified in ISO/IEC 7498-1:1994 (second edition).

The profile ISO/IEC ISP 12070 was planned as an International Standardized Profile in multiple parts. Each part would provide requirements that may be used to ensure a consistent approach when specifying the use of coded character sets in functional standards. Since activity on OSI has now come to a halt internationally, it is unlikely that that the remaining parts will be produced.

Within the set of character set standards there are two generic code structures, that defined by ISO/IEC 2022 for 7-bit and 8-bit transport mechanisms and that defined by the new ISO/IEC 10646 for a multiple-octet transport mechanism. It was the intention to cover both of these generic code structures as future parts were prepared. Part 1 is concerned with the ISO/IEC 2022 code structure. The requirements that it specifies apply specifically to Western Europe but they may be applicable also to other regions of the world.

6.1.12.3 *Tutorial guidance*

Part 1 of ISO/IEC ISP 12070 is applicable to use of ISO/IEC 2022 code structure in the following cases:

- in the ASN.1 type GeneralString;
- in the ASN.1 type GraphicString;
- in the ASN.1 type TeletexString;
- outside of the ASN.1 character string types, such as when a character string is embedded in the ASN.1 OCTET STRING type.

Abstract Syntax Notation One (ASN.1) forms part of the Presentation Layer of the OSI model. It is specified in the four-part standard ISO/IEC 8824, the current edition of which was published in 1995, replacing the second edition of 1990 that was a one-part standard. The ASN.1 standard defines a number of character string types, of which those listed above are a selection.

The last case in the list is particularly relevant for character based data streams and for this reason ISO/IEC ISP 12070-1 is included in this annex. It contains a number of specific conformance requirements which go a long way towards supporting implementation interoperability. These could be used as a checklist for developers but also as requirements to be referenced during the procurement process.

6.2 International Registers

6.2.1 ISO 2375 Register (International Register of Coded Character Sets to be used with Escape Sequences)

6.2.1.1 *Current edition*

ISO 2375 Register of International Register of Coded Character Sets to be used with Escape Sequences.

This register is updated as additional entries are approved in accordance with the procedures of ISO 2375. It does not have discrete editions. It is available from the Registration Authority:

> IPSJ/ITSCJ (Information Processing Society of Japan/Information Technology Standards Commission of Japan), Room 308-3, Kikai-Shinko-Kaikan Bldg., 3-5-8, Shiba-Koen, Minato-ku, Tokyo 105 JAPAN

The register used to be available free of charge in paper form. However, the register is now available in a much more convenient form electronically on the Web.

6.2.1.2 *Description*

This register contains the specifications of all coded character sets and control functions that can be brought into use by means of escape sequences. Many, but not all, of the coded character sets specified in the register are taken from national and international standards.

Each register entry is assigned a registration number that provides an unambiguous method for referencing that entry. It is specified in ISO 2375 that the entry with registration number *nn* should be referenced as ISO-IR *nn*.

6.2.1.3 *Tutorial guidance*

The ISO 2375 Register is divided into the following sections and subsections:

- Graphic character sets
    - 94-character sets, i.e. 94-position single-byte sets
    - 96-character sets, i.e. 96-position single-byte sets
    - Multiple-byte sets, which may be either 94-position or 96-position sets
- Control character sets
    - C0 sets of control characters
    - C1 sets of control characters
- Single control functions
- Coding systems different from that of ISO/IEC 2022
    - Coding systems with a standard return
    - Coding systems without a standard return

The registration numbers and escape sequences for a number of register entries are given elsewhere in this annex. These entries may be found as follows:

- For 94-character single-byte sets that provide national and other versions of ISO/IEC 646 (variants of ASCII), see 6.1.1, ISO/IEC 646.

- For 96-character single-byte sets that supplement ISO-IR 6 (ASCII) to provide characters of the Greek, Cyrillic, Arabic and Hebrew scripts and additional Latin characters, see the guide to ISO/IEC 10367 in 6.1.9. This reference includes the character sets of ISO/IEC 6937 and ISO/IEC 8859.

- For two-byte sets from Chinese, Japanese and Korean national standards that provide coding for ideographic scripts, see 5.2.2.3.

- For single control functions for code extension, see 5.3.4.1.

Register entries which specify C0 and C1 sets of control functions include:

- C0 sets:

    - ISO-IR 1 (ESC 02/01 04/00) : C0 set of ISO/IEC 6429

    - ISO-IR 104 (ESC 02/01 04/07) : Minimum C0 set for ISO/IEC 4873

      This set comprises only the character ESCAPE (ESC).

    - ISO-IR 106 (ESC 02/01 04/05) : Teletex primary set of Control Functions of CCITT Recommendation T.61.

      This set comprises the control characters BACKSPACE (BS), LINE FEED (LF), FORM FEED (FF), CARRIAGE RETURN (CR), LOCKING SHIFT ZERO (LS0), LOCKING SHIFT ONE (LS1), SINGLE-SHIFT TWO (SS2), SINGLE-SHIFT THREE (SS3), SUBSTITUTE CHARACTER (SUB) and ESCAPE (ESC) only.

- C1 sets:

    - ISO-IR 77 (ESC 02/02 04/03) : C1 set of ISO 6429:1983

      The C1 set of ISO/IEC 6429:1992 is designated as the first revision of ISO-IR 77 by means of the escape sequences ESC 02/06 04/00 ESC 02/02 04/03; see designation of sets of control functions in the section on control functions.

    - ISO-IR 105 (ESC 02/02 04/07): Minimum C1 set for ISO/IEC 4873

      This set comprises only the control characters SINGLE-SHIFT TWO (SS2) and SINGLE-SHIFT THREE (SS3).

    - ISO-IR 107 (ESC 02/02 04/08) : Teletex supplementary set of Control Functions of CCITT Recommendation T.61.

      This set comprises the control functions PARTIAL LINE UP (PLU), PARTIAL LINE DOWN (PLD) and CONTROL SEQUENCE INTRODUCER (CSI) only.

The escape sequences listed designate these sets as the C0 or C1 code element, as appropriate.

6.3 European Standards

6.3.1 EN 1922

6.3.1.1 *Current edition*

EN 1922:1998, Information technology – Character repertoire and coding for interworking with telex services (first edition) .

6.3.1.2 *Description*

European Standard EN 1922 is intended to be used with, and identified within, other European functional standards that specify strings of coded characters for interchange of coded information between information processing systems. It describes the graphic character repertoire and control functions relevant for information interchange via Telex network equipment.

European Standard EN 1922 is a revision of European Pre-standard ENV 41504:1990.

6.3.1.3 *Tutorial guidance*

EN 1922 specifies two repertoires, one each for the Latin and Greek scripts. The Latin repertoire is suitable for coding by means of the 5-bit Baudot code described in the historical background in this annex and which is more properly described as CCITT International Telegraphic Alphabet No.2. The Greek repertoire is similarly suitable for coding by means of the 5-bit code specified in the Hellenic national standard ELOT 1095:1989.

EN 1922 also specifies a transformation procedure for interconverting between data coded according to these 5-bit codes and data coded in an 8-bit code according to one of the options of EN 1923.

6.3.2 EN 1923

6.3.2.1 *Current edition*

EN 1923 (to be published), European character repertoires and their coding – 8 bit single byte coding (first edition).

6.3.2.2 *Description*

European Standard EN 1923 specifies the graphic character repertoires, and their coding, which are available for use for information interchange between information processing systems and for use within such systems, in the scripts that are commonly used by the members of CEN and the institutions of the European Union and the European Free Trade Association.

EN 1923 is a successor to three European Pre-standards, namely ENV 41503, ENV 41505 and ENV 41508.

6.3.2.3 *Tutorial guidance*

EN 1923 specifies a number of repertoires of characters selected from the Latin, Greek and Cyrillic scripts and a further repertoire of symbols. Each repertoire is assigned a name and a mnemonic. The names and mnemonics are:

- The Invariant-Latin repertoire (IVL), which is a subset of:
- The Initial-Latin repertoire (IL), which is a subset of:
- The Basic-Latin repertoire (BL), which is a subset of:
- The Full-Latin-8 repertoire (FL8).
- The Basic-Greek repertoire (BG).
- The Basic-Cyrillic repertoire (BC).
- The Symbols-1 repertoire (BS).

A number of options are specified within the standard, each of which identifies a requirement to support one or more of the above repertoires. When 8-bit single-byte coding is used to support of any of the identified options, it is required to conform to ISO/IEC 4873 at one of its three levels of implementation.