ISO/IEC TR 15285

Information technology —

An operational model for characters and glyphs

Technologies de l’information —
Modèle pour l’utilisation de caractères graphiques et de glyphes

Version: *29 June, 1998*

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The main task of a technical committee is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

ISO/IEC TR 15285, which is a Technical Report of type 3, was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 2, *Coded character sets* and Subcommittee SC 18, *Document processing and related communication* (which has since been reorganized into SC 33, *Distributed application services)*.

Introduction

People interpret the meaning of a written sentence by the shapes of the characters contained in it. Reduced to the character level, people consider the information content of a character inseparable from its printed image. Information technology, in contrast, makes a distinction between the concepts of a character's meaning (the information content) and its shape (the presentation image). Information technology uses the term *character* (or *coded character*) for the information content, and the term *glyph* for the presentation image. A conflict exists because people consider characters and glyphs equivalent. Moreover, this conflict has led to misunderstanding and confusion. This Technical Report provides a framework for relating characters and glyphs to resolve the conflict because successful processing and printing of character information on computers requires an understanding of the appropriate use of characters and glyphs.

Historically, ISO/IEC JTC 1/SC 2 has had responsibility for the development of coded character set standards such as ISO/IEC 10646 for the digital representation of letters, ideographs, digits, symbols, etc. ISO/IEC JTC 1/SC 18 has had responsibility for the development of standards for document processing, which presents the characters coded by SC 2. SC 18 standards include the font standard, ISO/IEC 9541, and the glyph registration standard, ISO/IEC 10036. The Association for Font Information Interchange (AFII) maintains the 10036 glyph registry on behalf of ISO.

This Technical Report is written for a reader who is familiar with the work of SC 2 and SC 18. Readers without this background should first read Annex B, “Characters”, and Annex C, “Glyphs”.

This edition of the Technical Report does not fully develop the complex issues associated with the Chinese, Japanese, Korean, and Vietnamese ideographic characters used in East Asia. In addition, although it discusses the process of rendering digital character information for display and printing, it avoids discussing the inverse process of character recognition (i.e. converting printed text into character information in the computer).

Information technology — An operational model for characters and glyphs

1 Scope

The purpose of this Technical Report is to provide a general framework for discussing characters and glyphs. The framework is applicable to a variety of coded character sets and glyph-identification schemes. For illustration, this Technical Report uses examples from characters coded in ISO/IEC 10646 and glyphs registered according to ISO/IEC 10036.

This Technical Report:

— differentiates between coded characters and registered glyphs

— identifies the domain of use of coded characters and glyph identifiers

— provides a conceptual framework for the formatting and presentation of coded character data using glyph identifiers and glyph representations

This Technical Report describes idealized principles that were not completely followed in coding characters for ISO/IEC 10646 and in registering glyphs according to ISO/IEC 10036. The fact that ISO/IEC 10646, ISO/IEC 10036, and other standards do not completely follow the principles in the model does not invalidate the model and does not diminish the utility of having the model.

2 References

ISO/IEC 9541-1: 1991, *Information technology — Font information interchange — Part 1: Architecture.*

ISO/IEC 10036: 1996, *Information technology — Font information interchange — Procedures for registration of font-related identifiers.*

ISO/IEC 10180: 1995, *Information technology — Processing languages — Standard Page Description Language (SPDL).*

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

3 Definitions

For the purpose of this Technical Report, the following definitions apply. The definitions have been extracted from the ISO/IEC 9541-1: 1991 and ISO/IEC 10646-1: 1993 standards.

3.1 character: A member of a set of elements used for the organisation, control, or representation of data. (ISO/IEC 10646-1: 1993)

3.2 coded character set: A set of unambiguous rules that establishes a character set and the relationship between the characters of the set and their coded representation. (ISO/IEC 10646-1: 1993)

3.3 font: A collection of glyph images having the same basic design, e.g. Courier Bold Oblique. (ISO/IEC 9541-1: 1991)

3.4 font resource: A collection of glyph representations together with descriptive and font metric information which are relevant to the collection of glyph representations as a whole. (ISO/IEC 9541-1: 1991)

3.5 glyph: A recognizable abstract graphic symbol which is independent of any specific design. (ISO/IEC 9541-1: 1991)

3.6 glyph collection: An identified set of glyphs. (ISO/IEC 9541-1: 1991)

3.7 glyph image: An image of a glyph, as obtained from a glyph representation displayed on a presentation surface.

(ISO/IEC 9541-1: 1991) [See the definition of *graphic symbol*.]

3.8 glyph metrics: The set of information in a glyph representation used for defining the dimensions and positioning of the glyph shape. (ISO/IEC 9541-1: 1991)

3.9 glyph representation: The glyph shape and glyph metrics associated with a specific glyph in a font resource. (ISO/IEC 9541-1: 1991)

3.10 glyph shape: The set of information in a glyph representation used for defining the shape which represents the glyph. (ISO/IEC 9541-1: 1991)

3.11 graphic character: A character, other than a control function, that has a visual representation normally handwritten, printed, or displayed. (ISO/IEC 10646-1: 1993)

3.12 graphic symbol: The visual representation of a graphic character or of a composite sequence. (ISO/IEC 10646-1: 1993) [See the definition of *glyph image*.]

3.13 presentation [of a graphic symbol]: The process of writing, printing, or displaying a graphic symbol. (ISO/IEC 10646-1: 1993)

3.14 presentation form: In the presentation of some scripts, a form of a graphic symbol representing a character that depends on the position of the character relative to other characters. (ISO/IEC 10646-1: 1993)

3.15 presentation surface: A virtual representation of a presentation medium (page, graphic display, etc.) maintained by the presentation process, on which all glyph shapes are to be imaged. (ISO/IEC 9541-1: 1991)

3.16 repertoire: A specified set of characters that are represented in a coded character set. (ISO/IEC 10646-1: 1993)

4 Character and glyph distinctions

The character and glyph definitions in clause 3, which were taken from ISO/IEC 10646 and ISO/IEC 9541, were developed independently and contain terminology that requires explanation.

In information technology, *characters* are abstract information elements in the domain of coding for data representation, and in particular data interchange. Coded character set standards assign numeric values, character names, and representative (sample) images to each character contained in a coded character set. Typically a character is given a name, which also serves to differentiate it from the other characters of the coded character set. The precise semantics and appearance of the information elements in any given implementation are not defined by those standards for coded character sets. This apparent lack of definition is not considered to be a defect in the standards. Recognizing that the information may be acted upon (deciphered, sorted, transformed, formatted, archived, presented, etc.) by many different application processes during its lifetime, standards for coded character sets are defined as a basis for information interchange.

In information technology, *glyphs* are abstract presentation elements in the domain of presentation processing. The ISO/IEC 10036 standard for glyph registration defines the process for assigning glyph identifiers, glyph descriptions, and representative (sample) images to each glyph submitted for registration. The precise usage and appearance of these presentation elements in any implemented font resource is not defined by those glyph registration activities. As with the coded character set standards, this apparent lack of definition is not considered to be a defect in the standards. Glyph identifiers are unambiguously assigned as a basis for tagging presentation elements in and among interchanged font resources, recognizing that the font-specific design information may vary from one font resource to another.

Characters and *glyphs* are closely related, with many attributes in common and yet with distinctions that make it essential that they be managed in information processing as separate entities. The ISO/IEC 10646 standard recognizes the distinction between characters and their visual representation by defining the term, *graphic symbol*. The *graphic symbol* of SC 2 standards and the *glyph image* of SC 18 standards represent equivalent concepts. However, *glyph* and its associated ISO/IEC 9541 terminology are preferred when referring to presentation and presentation processing.

The historical association of characters and glyphs has resulted in character sets maintaining distinctions that cannot be founded on distinctions in meaning, but only distinctions in shape. Similarly, the glyph registration authority and the SC 18 font resource model have made use of criteria based on meaning to abstract potential distinctions in shape. In practice, ISO/IEC 10646 contains characters that appear to be instances of glyphs, while the glyph registry prescribed by ISO/IEC 10036 contains glyphs that appear to be designated as abstract characters. In both cases, the ideal nature of characters and glyphs has been compromised to a degree. For example, in ISO/IEC 10646-1, SC 2 coded the “ﬁ” glyph into the character U+FB01 LATIN SMALL LIGATURE FI “ﬁ” for round-trip integrity with other standards.[1] (See Annex B.5 The “round-trip rule”.) Also, the JTC 1 Registration Authority (AFII) for ISO/IEC 10036 could have registered the same glyph identifier for the “A” glyph and used it for the U+0041 LATIN CAPITAL LETTER A “A” character, for the U+0391 GREEK CAPITAL LETTER ALPHA “A” character, and the U+0410 CYRILLIC CAPITAL LETTER A “A” character. However, AFII instead registered three glyph identifiers.

Within the realm of information technology, an ideal characterization of characters and glyphs and their relationship may be stated as follows:

— A *character* conveys distinctions in meaning or sounds. A character has no intrinsic appearance.

— A *glyph* conveys distinctions in form or appearance. A glyph has no intrinsic meaning.

— One or more characters may be depicted by no, one, or multiple glyph representations (instances of an abstract glyph) in a way that may depend on the context.

5 Operational model

5.1 Character and glyph domains

Character information has two primary domains as illustrated in Figure 1. The first pertains to the *processing* of the content, i.e. the meaning or phonetic value of the character information. This is depicted on the left side of the figure. The second pertains to the *presentation* of the content of the character information. This is depicted on the right side of the figure.[2] Each domain places different requirements on the representation of the character information. For example, searching for character information in a database and sorting records containing character information entail different requirements than those found in presenting characters on paper. The former processes are primarily concerned with the content of data and have little or no concern about the appearance that the data may take.

[1] This Technical Report describes a character in terms of its 10646 code position (U+FB01), its 10646 name (LATIN SMALL LIGATURE FI), and illustrates it with a representative glyph in quotation marks (“ﬁ”).

[2] ISO/IEC 6429 also depicts a 2-layer structure. For ISO/IEC 6429, the data layer could use characters, and the presentation layer could use glyphs to present the characters in the data layer.

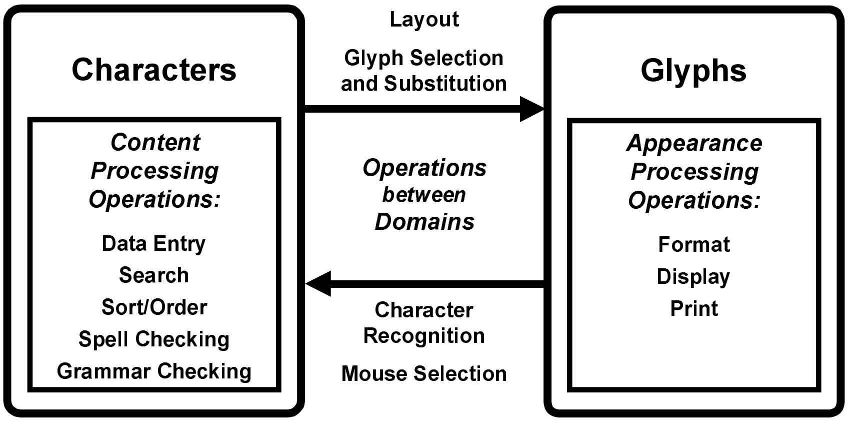

Figure 1 — Character and glyph domains

On the other hand, a composition and layout process has little concern for the content of data, but great concern about its appearance. In general, processing of character information in the content domain is independent of font resources, whereas processing in the presentation domain is strongly dependent on the font resource used for the presentation of the character information. However, processes that perform transformations from one domain to the other are aware of both the content and appearance of characters. For example, a character recognition process converts images into coded characters. Also, a paragraph-level hyphenation process is an example of a layout process that requires content information.

It is not possible, in general, to code data in such a way as to optimize one process without reducing the performance of other processes. Even within the content domain, the nature of the character coding employed for textual data affects the type or types of processing to be performed on the data; no single coding can optimize more than a few such potential processes. Given this situation, the best solution is to formulate an independent, logical character coding that, when necessary, can be transformed into another coding more amenable to the processing required. For example, in the case of searching, character data is often recast into specific forms that facilitate quick searches. For sorting, a specially created sort key is required. In addition, because ISO/IEC 10646 contains glyph-like characters, it is expected that implementations may choose to canonicalize or normalize such characters by translating them to normative characters. A presentation subsystem that employs such a technique may require that character data be normalized prior to presentation.

The recognition that two separate domains of processing are commonly applied to character-based information leads to a conclusion that two primary forms of this information are needed:

1. a content-oriented form that is amenable to immediate content-based processes and that can be easily converted to and from other optimized forms

2. an appearance-oriented form that facilitates imaging of content

These are, respectively, the character-based form and the glyph-based form. Failure to recognize this distinction between the character domain and the glyph domain has led to the development of inconsistent standards and inconsistent systems that lack functional separation of the two domains.

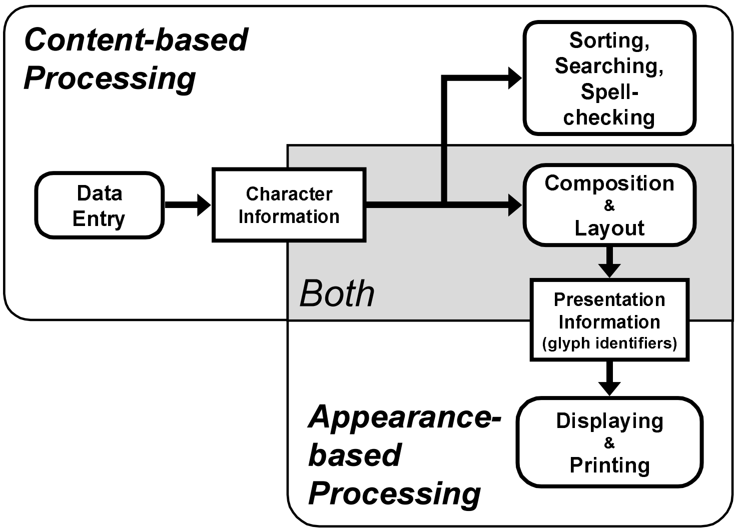
Content-based Processing
Sorting, Searching, Spell-checking
Data Entry
Character Information
Composition & Layout
Both
Presentation Information (glyph identifiers)
Appearance-based Processing
Displaying & Printing

Figure 2 — Composition, layout, and presentation

5.2 Composition, layout, and presentation

As depicted in Figure 2 on the following page, the composition and layout process (for glyph selection and positioning) spans both processing domains. If attention is restricted to the text portion of this process, the presentation of character-based information requires three primary operations:

— selecting the glyph representations needed to display character data

— positioning the glyph shapes on the presentation surface

— imaging the glyph shapes

Glyph selection is the process of selecting (possibly through several iterations) the most appropriate glyph identifier or combination of glyph identifiers to render a coded character or composite sequence of coded characters. Coded characters and their associated implicit or explicit formatting information (e.g. specification of the font and its size) represent the primary inputs to composition and layout processing, and glyph identifiers (or the associated glyph metrics and glyph shapes) represent the primary output from composition and layout processing. The degree of glyph selection sophistication varies widely among existing standards and implementations.

The relationship between coded characters and glyph identifiers may be one-to-one, one-to-many, many-to-one, or many-to-many.[3] This is particularly true for ISO/IEC 10646 implementation level 3, which uses combining characters. In its fully general form, the relationship is a context-sensitive M-to-N mapping where $M > 0$, $N \geq 0$. For some characters in ISO/IEC 10646-1, no glyph (N=0) can be defined, for example, the U+FEFF ZERO WIDTH NO-BREAK SPACE character.

The SC 18 document-processing model separates the glyph selection and layout operations from the operation of imaging the glyph shape to permit document interchange between the processes. Glyph selection and positioning are part of the

[3] The necessity for mapping characters to glyphs (glyph selection), not its complexity, is one of the motivations for developing this operational model for characters and glyphs.

composition and layout process, whereas imaging the glyph shape is part of the presentation process. The result of composition and layout is a *final-form document*, which contains font identifiers, glyph identifiers, and coordinate positions, along with either references to font resources or the actual font resources themselves. Such a document form contains all the necessary information required to present the formatted document on some presentation medium. An example of such a final form document is an SPDL (ISO/IEC 10180) document instance.

An important aspect of this document-processing model is that it begins with coded-character data as its input and produces either glyph-based data or directly imaged glyph shapes as its output. That is, it incorporates a transformation from a coded-character representation of a document's *content* to a glyph-based coding of a document's *appearance*. The latter may only be visible to the internal mechanisms of an operating system or a user-interface subsystem in the case that the result is directly imaged for presentation. However, even these systems frequently support some form of output that contains the glyph-based final form of the document.

6 Glyph selection

While some earlier formatting systems assume a one-to-one correspondence between characters and glyphs, this is inadequate for many applications and scripts. Many contemporary composition and layout systems support more complex glyph-selection processes that provide for the representation of sequences of multiple character codes by a single glyph or by the use of sequences of glyphs to represent certain characters. In general, glyph selection needs to be based on style information and context as well as on the character data itself. For example, consider the following:

— When the U+0022 QUOTATION MARK “"” character is encountered, a composition and layout process may need to determine whether it begins or ends a quotation and then choose either an opening or closing quotation mark glyph (““” or “””) as appropriate. In addition, the process may select glyphs depending on the language of the text being formatted (or the formatting style specifications that apply to the content being formatted). For example, German text could use the “„” and ““” glyphs for quotation marks; and French text, the “«” and “»” glyphs.

— When the U+002D HYPHEN-MINUS “-” character is encountered, a composition and layout process may have to determine if it is used in a math formula, as a separator between figures (digits), as a separator between words, or as a separator between syllables. Depending on which context applies, it will select a minus sign, a figure dash, a quotation dash, or a hyphen dash (or possibly a hyphen point) glyph to display the character.

NOTE: Because the ISO/IEC 10646 repertoire includes the necessary characters, some applications resolve quotation marks and the hyphen-minus illustrated in the previous two points by converting to the appropriate 10646 *characters* as they are input rather than selecting the appropriate glyphs for presentation.

— When a parenthesis or square bracket character is encountered in a document being formatted in vertical lines (e.g. with East Asian ideographs), a composition and layout process may need to choose a vertical variant glyph form of the parenthesis or square bracket. It may also perform a similar selection for certain other characters such as U+30FC KATAKANA-HIRAGANA PROLONGED SOUND MARK “ー”, U+2014 EM DASH “—”, U+2025 TWO DOT LEADER “. .”, etc.

— When an Arabic letter is encountered in an Arabic, Farsi, Urdu, etc. document, then, if the Arabic style being used to display the text is of the Simplified Naskh type, a composition and layout process may have to choose an isolated, initial, medial, or final glyph form for the given letter according to its context in the document. For example, glyphs for U+0647 ARABIC LETTER HEH “ه” are shown in Figure 3.

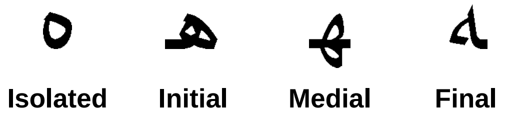

Figure 3 — Glyphs for ARABIC LETTER HEH

— In addition, Arabic typography makes extensive use of ligatures. For example, Figure 4 shows the isolated forms of U+0627 ARABIC LETTER ALEF “ا” and U+0644 ARABIC LETTER LAM “ل”, and then the two ligature forms used when Lam is followed by Alef.

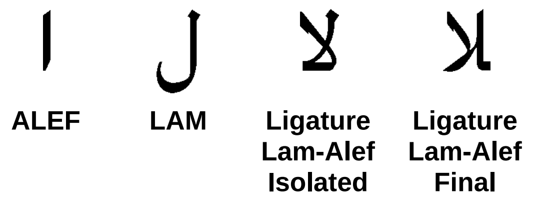

Figure 4 — Two example ligatures in an Arabic font

— When a U+0930 DEVANAGARI LETTER RA “र” is encountered in a Hindi, Marathi, Sanskrit, etc. document, a composition and layout process may have to determine whether a subscript, superscript, half (“eyelash”), or full form glyph is required according to context. If a subscript form is required, a composition and layout process may have to choose from one of a number of possible subscript forms depending on the glyph to which it is to be attached. Figure 5 shows an example of this.

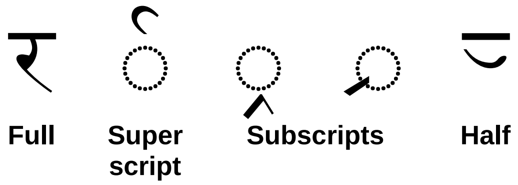

Figure 5 — Glyphs for DEVANAGARI LETTER RA

The process of glyph selection is sometimes implemented as a separate part of composition and layout because many of the choices required to determine an appropriate glyph are based solely on (1) the context of a character within a document, (2) the style specifications that apply to a given character, or (3) a combination of the context and style specification. All of the choices required for the examples shown above fall into one of these categories. However, in general, glyph selection can only be made as an integral part of the entire composition and layout process. Consider the following:

— When hyphenating a line of text during composition, a composition and layout process may insert a hyphen glyph form at the end of line if the line is broken at a hyphenation point.

— If hyphenating a German text between the letters “c” and “k”, a composition and layout process may replace the “c” with a “k”.

— If during the composition of a German text, the character sequence “fff” is encountered, a composition and layout process may select two distinct (non-ligated) glyph forms for U+0066 LATIN SMALL LETTER F “f”. However, if the position for a hyphen (a hyphen point) should occur before the last “f”, i.e. at “ff-f”, then a composition and layout process may select an ff ligature glyph “ﬀ”, followed by a hyphen (on the first line), and begin the subsequent line with a normal glyph for the third and final “f”.

— A composition and layout process may select small cap glyph forms for the first line of a paragraph of Roman text.

- A composition and layout process may select a swash glyph form for the first and last character of each line of a paragraph.

- A composition and layout process may select one of a number of possible variant glyph forms for certain Arabic letters depending on whether more or less space is available for composing a line of Arabic text.

- When justifying a line of Arabic text, a composition and layout process may start by selecting ligature glyph forms that consume the smallest amount of linear space in a line, and then sequentially replace these ligatures with component ligatures or component non-ligature glyphs such that more linear line space is consumed up to the required line measure. Alternatively, a composition and layout process may start justification by selecting no ligatures and then sequentially select ligatures that consume a smaller amount of linear space until the desired line measure is achieved or until an inter-word space stretch threshold is reached (i.e. a point at which inter-word spaces can be stretched to justify the line to the desired measure).

In summary, the glyph-selection process is primarily applicable to behavior occurring at the end or beginning of individual lines of text, or within the context of justifying or altering the measure of a given line during line composition. A system supporting the capabilities illustrated in the preceding examples must include glyph selection as an integral part of the composition and layout process.

7 Summary

Here are the primary points of this technical report:

- Most people equate a character and its shape.

- This causes difficulties and misunderstanding because contemporary information technology distinguishes two related, but distinct, domains:

  - a processing domain that uses coded characters to represent the character’s meaning
  - a presentation domain that uses glyph identifiers to represent the character’s image

- Processes are available to convert between the two domains:

  - Presentation processing takes the coded-character data plus any formatting data plus font information to display and print character data.
  - A character recognition process scans images, analyzes the shapes, and outputs the coded characters that correspond to the shapes.

- Depending on the script and the particular font or fonts used, glyph selection can be straightforward or relatively complex.

  - It is straightforward when a one-to-one correspondence exists between the set of coded characters and the set of registered glyphs in a font.
  - The process is more complex when it must choose between several alternatives; for example, when a sequence of coded characters may be mapped into more than one sequence of glyphs in a font.

Annex A
Bibliography

1. ISO/IEC 646: 1991, *Information technology — ISO 7-bit coded character set for information interchange.*

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

3. ISO/IEC 6937: 1993, *Information technology — Coded graphic character sets for text communication – Latin alphabet.*

4. ISO/IEC 8859, *Information technology — 8-bit single-byte coded graphic character sets*
   — Part 1. Latin alphabet No. 1 (1987)
   — Part 2. Latin alphabet No. 2 (1987)
   — Part 3. Latin alphabet No. 3 (1988)
   — Part 4. Latin alphabet No. 4 (1988)
   — Part 5. Latin/Cyrillic alphabet (1988)
   — Part 6. Latin/Arabic alphabet (1987)
   — Part 7. Latin/Greek alphabet (1987)
   — Part 8. Latin/Hebrew alphabet (1988)
   — Part 9. Latin alphabet No. 5 (1989)
   — Part 10. Latin alphabet No. 6 (1993).

5. ISO/IEC 10367: 1991, *Information technology — Standardized coded graphic character sets for use in 8-bit codes.*

6. ISO/IEC 10538: 1991, *Information technology — Control functions for text communication.*

7. ANSI X3.4-1986, *American National Standard for Information Systems — Coded Character Sets — 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)*.

8. JIS X 0201-1976, Japanese Standards Association, *Jouhou koukan you fugou (Code for Information Interchange)*.

9. JIS X 0208-1990, Japanese Standards Association, *Jouhou koukan you kanji fugoukei (Code of the Japanese Graphic Character Set for Information Interchange)*.

10. Becker, Joseph D., “Multilingual Word Processing”, *Scientific American*, Vol. 251, No. 1, July, 1984, pp. 96-107.

11. Bringhurst, Robert, *The Elements of Typographic Style*, Hartley and Marks, Vancouver, 1996.

12. Hartmann, R. R. K. and Stork, F. C., *Dictionary of language and linguistics*, Applied Science Publishers Ltd., London, 1976.

13. Lofting, Peter, “The Perception of Character Entities in Unfamiliar Scripts”, unpublished paper, July, 1995.

14. The Unicode Consortium, *The Unicode Standard, Version 2.0*, Addison-Wesley, Reading, MA, 1996.

Annex B
Characters

B.1 Definition

Quoting from ISO/IEC 10646-1:1993, the definition of *character* accepted by SC 2 is:

> *A member of a set of elements used for the organisation, control, and representation of data.*

This definition asserts (1) that, in the context of the role of SC 2, a *character* is an element of a larger set, a *character set*; and (2) that a character is used to represent data or to organize and control data, or with a few cases, both. The division between *data characters* and *control characters* is usually specified by requiring the former to be *graphical characters*, i.e. characters with which some graphical form can be associated. A character is not generally found (or interpreted) in isolation, but appears as an element of a sequence (an array) of characters, i.e. a character string, and therefore is interpreted according to the context in which it appears.

After defining a character in this fashion, SC 2 defines character sets by enumerating a list of characters. Such characters are enumerated by assigning a unique name to each character, by specifying a unique code (the *code position*), and by depicting a representative image in a table (the *code table*). In general, this describes the entire formal content of any given SC 2 coded character set standard, although various standards sometimes augment their formal content with additional information, particularly information pertaining to characters that participate in control functions.

B.2 Character information

What SC 2 does not do—and this is perhaps the most important point of this annex—is formally define the data or units of information that graphic characters are supposed to represent; that is, no formal semantics are specified to assist in the task of interpreting the so-called data supposedly being represented by a character. Instead, SC 2 assumes that the semantics of a character is either (1) self-evident; or (2) subject to conventions adhered to by the user of the character, namely, the application.

In a small character set standard, such as ISO/IEC 646: 1991, the process of determining the information represented by each character is relatively straightforward and usually involves the invocation of self-evident knowledge. For example, the characters of ISO/IEC 646 that appear to be the letters of the modern English alphabet, and to which are assigned names that appear to be the names of the letters of this alphabet, are indeed usually assumed to represent none other than the English alphabet. However, this assumption is not supported by the formal definition of ISO/IEC 646. Nowhere in this standard does it specify that these characters actually represent information to be interpreted as letters of the English alphabet. Indeed, an application developer who happens to be Hawaiian may interpret these characters as representing the elements of the Hawaiian alphabet (plus a few extra letters not used by Hawaiian), or, a Japanese developer may interpret them as representing the elements of the Romaji form of written Japanese. In each case, the user of the standard is applying conventions that do not conflict with the standard itself, and that enable the user to employ the standard in a useful way. Other elements of ISO/IEC 646, such as the character assigned to positions 2/13 (U+002D HYPHEN-MINUS “-”) and 2/7 (U+0027 APOSTROPHE “ ' ”) are commonly given multiple interpretations depending on their use. For example, the latter character may be used as an apostrophe, as a single quote mark, or, in some transliteration systems, as standing for a glottal stop or a palatalized consonant. Since the standard does not specify which information the character represents, a user of the standard is free to choose. Once the number of characters in a standard is increased many times, such as the case with ISO/IEC 10646-1: 1993 where over 30,000 characters are defined, the potential for multiple usage conventions increases.

B.3 Example, the unit of information “one”

Consider for a moment the case with the unit of information meaning “one”. ISO/IEC 10646 not only codes a large number of characters that conceivably represent this unit of information, but also codes a number of characters that represent a particular form associated with this meaning. The characters that may be said to represent the unit of information designated by “one” are (at least):

U+0031	DIGIT ONE “1”
U+00B9	SUPERSCRIPT ONE “¹”
U+0661	ARABIC-INDIC DIGIT ONE “١”
U+06F1	EXTENDED ARABIC-INDIC DIGIT ONE “۱”
U+0967	DEVANAGARI DIGIT ONE “१”
U+09E7	BENGALI DIGIT ONE “১”
U+09F4	BENGALI CURRENCY NUMERATOR ONE “৴”
U+0A67	GURMUKHI DIGIT ONE “੧”
U+0AE7	GUJARATI DIGIT ONE “૧”
U+0B67	ORIYA DIGIT ONE “୧”
U+0BE7	TAMIL DIGIT ONE “௧”
U+0C67	TELUGU DIGIT ONE “౧”
U+0CE7	KANNADA DIGIT ONE “೧”
U+0D67	MALAYALAM DIGIT ONE “൧”
U+0E51	THAI DIGIT ONE “๑”
U+0ED1	LAO DIGIT ONE “໑”
U+2081	SUBSCRIPT ONE “₁”
U+215F	FRACTION NUMERATOR ONE “⅟”
U+2160	ROMAN NUMERAL ONE “Ⅰ”
U+2170	SMALL ROMAN NUMERAL ONE “ⅰ”
U+2460	CIRCLED DIGIT ONE “①”
U+2474	PARENTHESIZED DIGIT ONE “⑴”
U+2488	DIGIT ONE FULL STOP “⒈”
U+2776	DINGBAT NEGATIVE CIRCLED DIGIT ONE “❶”
U+2780	DINGBAT CIRCLED SANS-SERIF DIGIT ONE “➀”
U+278A	DINGBAT NEGATIVE CIRCLED SANS-SERIF DIGIT ONE “➊”
U+3021	HANGZHOU NUMERAL ONE “〡”
U+3192	IDEOGRAPHIC ANNOTATION ONE MARK “㆒”
U+3220	PARENTHESIZED IDEOGRAPH ONE “㈠”
U+3280	CIRCLED IDEOGRAPH ONE “㊀”
U+4E00	CJK UNIFIED IDEOGRAPH-4E00 “一”
U+58F9	CJK UNIFIED IDEOGRAPH-58F9 “壹”
U+FF11	FULLWIDTH DIGIT ONE “１”

Of these characters, the following are merely size or position variants of a single form:

U+0031	DIGIT ONE “1”
U+00B9	SUPERSCRIPT ONE “¹”
U+2081	SUBSCRIPT ONE “₁”
U+FF11	FULLWIDTH DIGIT ONE “１”

The following are various adorned variants of this form:

U+215F	FRACTION NUMERATOR ONE “⅟”
U+2460	CIRCLED DIGIT ONE “①”
U+2474	PARENTHESIZED DIGIT ONE “⑴”
U+2488	DIGIT ONE FULL STOP “⒈”
U+2776	DINGBAT NEGATIVE CIRCLED DIGIT ONE “❶”
U+2780	DINGBAT CIRCLED SANS-SERIF DIGIT ONE “➀”
U+278A	DINGBAT NEGATIVE CIRCLED SANS-SERIF DIGIT ONE “➊”

The remaining characters, although all represent the concept “one”, employ different forms depending on the script with which they are associated. However, one could argue that a number of these forms are really different instances of a single form from which they are historically derived, namely the Indic-script forms of “one”:

U+0661	ARABIC-INDIC DIGIT ONE “١”
U+06F1	EXTENDED ARABIC-INDIC DIGIT ONE “۱”
U+0967	DEVANAGARI DIGIT ONE “१”
U+09E7	BENGALI DIGIT ONE “১”
U+0A67	GURMUKHI DIGIT ONE “੧”
U+0AE7	GUJARATI DIGIT ONE “૧”
U+0B67	ORIYA DIGIT ONE “୧”
U+0BE7	TAMIL DIGIT ONE “௧”
U+0C67	TELUGU DIGIT ONE “౧”
U+0CE7	KANNADA DIGIT ONE “೧”
U+0D67	MALAYALAM DIGIT ONE “൧”
U+0E51	THAI DIGIT ONE “๑”
U+0ED1	LAO DIGIT ONE “໑”
U+3021	HANGZHOU NUMERAL ONE “〡”
U+3192	IDEOGRAPHIC ANNOTATION ONE MARK “㆒”
U+3220	PARENTHESIZED IDEOGRAPH ONE “㈠”
U+3280	CIRCLED IDEOGRAPH ONE “㊀”
U+4E00	CJK UNIFIED IDEOGRAPH-4E00 “一”
U+58F9	CJK UNIFIED IDEOGRAPH-58F9 “壹”

This example clearly shows that the designers of this character set did not start with individual units of information and assign each such unit to a unique character; furthermore, it is also clear that the designers did not start with individual forms and assign each to a unique character. Rather, a combination of forms and variations of a single form, all signifying the idea “one”, were included as distinct characters.

To gain an understanding for the distinction between characters and glyphs, consider that the following characters could have easily been unified into a single character that would be displayed using one of four glyphs.

U+0031	DIGIT ONE “1”
U+00B9	SUPERSCRIPT ONE “^{1}”
U+2081	SUBSCRIPT ONE “$_1$”
U+FF11	FULLWIDTH DIGIT ONE “１”

These four characters can be considered as instances of one character that takes on slightly different forms depending on usage. In this case, usage or style alone would govern the form chosen to depict a single abstract character. In the case of a form used as the numerator of a fraction, the appropriate glyph could be determined based on the local context of the character, assuming for a moment that a character such as a U+0031 DIGIT ONE “1” is followed by a U+2044 FRACTION SLASH “⁄”. In the remaining cases, the character’s immediate context would not be sufficient, but would require additional information be supplied such as style information that would govern the appearance of a character when displayed. In either case, the process of depicting a given character may require the selection of one of a number of possible glyphs, each of which may serve (in different cases) to present the image of a character.

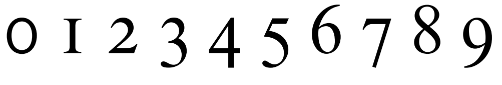

Figure 6 — Old Style Figures

Notice that certain other possible forms of a “one” are, in fact, not found in this standard as characters. For example, many high quality font collections supply a collection of forms for the Arabic numerals known as *old style figures* shown in Figure 6. Were the *old style figures* included as characters, the OLD STYLE FIGURE DIGIT ONE “I” could have been added to 10646.

B.4 Considerations for deciding the repertoire of a coded character set

Various arguments are possible for defending the inclusion or exclusion of a particular form as a possible graphic character in a repertoire. In many cases, the criterion for either inclusion or exclusion has not been articulated, but is based on informal opinion about appropriateness. Justifying why certain forms were coded into ISO/IEC 10646-1: 1993, and why others were not, is beyond the scope of this Technical Report. However, with respect to coding glyphs versus characters, the objective is to code characters that represent different information. To meet this objective, three important considerations should be applied.[4]

1. *Same shape/different meanings*

   Does one shape have multiple meanings (semantics)?

   Some shapes will be the same, or nearly the same, but have different meanings or different semantics. An example of this is that in many sans-serif fonts the glyph “I” is used for both the U+0049 LATIN CAPITAL LETTER I “I” and the U+006C LATIN SMALL LETTER L “l”. Similarly, for years many typewriters lacked a key for the U+0031 DIGIT ONE “1” and people were taught to type the U+006C LATIN SMALL LETTER L “l” instead. Later when people switched from typewriters to computers, this practice failed and people had to relearn to type the digit one “**1**” instead of the letter “**l**”.

[4] Peter Lofting, “The Perception of Character Entities in Unfamiliar Scripts”.

2. *Different shapes/same meaning*

   Do two or more shapes imply the same meaning (semantics)?

   Shape differences may be font design differences or glyph rendering differences. Examples of font design differences (for which the different shapes would have the same glyph identifier in the ISO/IEC 10036 glyph register) are the “a” and “*a*” glyph variations of the U+0061 LATIN SMALL LETTER A “a”. Examples of glyph rendering differences (for which the different shapes would have different glyph identifiers) are the Arabic letters and corresponding initial, medial, and final presentation forms. Figure 3 illustrates this concept. It is important to discern small differences and determine when they are merely embellishments and when they change the meaning. For example, the shape of the U+0428 CYRILLIC CAPITAL LETTER SHA “Ш” differs very little from the shape of the U+0429 CYRILLIC CAPITAL LETTER SHCHA “Щ”, yet they are different letters.

3. *Compatibility*

   Is the shape needed for migration of, and coexistence with, text coded using an older coded character set?

   In practice, the need for compatibility with existing coded character sets frequently overrides the second consideration. Examples of this are found in ISO/IEC 10646-1: 1993. The next clause describes an important compatibility criterion, the “round-trip rule”.

These considerations should be used to help decide which forms to include in a *new* repertoire to be coded. Although the considerations are easy to state, obtaining definitive answers requires considerable effort, e.g. to consult with experts and native users, who are normally unaware of information technology and not concerned with such details.

B.5 The “round-trip rule”

In the case of ISO/IEC 10646, an informal criterion (known as the “round-trip rule”) for the inclusion of a character can be phrased as follows:

> *If a form is included as a character in any of the character sets from which ISO/IEC 10646 is derived, then that form shall be included as a character in ISO/IEC 10646 such that distinctions among characters in the source character set are maintained as distinctions in ISO/IEC 10646.*

This criterion was defined such that the elements of two source character sets could be unified with each other (e.g. the ideographic characters in the Chinese, Japanese, and Korean national standards) while at the same time guaranteeing that distinctions within a source character set would be maintained. The latter was required to guarantee that no loss of information would occur when translating from one of the source character sets to 10646 and then back to the original character set.

Certain characters that might have been unified in 10646 were not unified because of the round-trip rule. For instance, U+00B9 SUPERSCRIPT ONE “^{1}” was not unified with U+0031 DIGIT ONE “1” because ISO 8859-1: 1987, a source character set for 10646, includes these two forms as distinct characters. Most of the instances of formal entities within 10646 that could have been unified were likewise distinct characters in some source character set, or, in some special instances, distinct characters in certain unions of character sets, e.g. the union of 7-bit ASCII (ANSI X3.4-1986), JIS X 0201-1976, and JIS X 0208-1990 as employed in Shift JIS coding in Japan.

Annex C
Glyphs

C.1. Definition

SC 18 defines a *glyph* as:

> *An abstract identified graphical symbol independent of any actual image.*

Two aspects of this definition are important to consider: (1) a glyph is identifiable; and (2) a glyph is an abstraction of an actual image. The notion of identification is closely tied to the use of a glyph. In the SC 18 model of font resources, articulated by ISO/IEC 9541, ISO/IEC 10180 (SPDL), et al., each element of a font resource must be capable of identification. This identification facilitates the unique selection of the representation of a glyph from a font resource, and the interchange of such identifications embedded in the formatted, final form of a document, e.g. an ISO/IEC 10180 file. The definition of a final-form document specifies that all composition and layout operations have already taken place and, in particular, that the selection of the glyphs that will be employed to depict character data has already occurred. The business of defining identifiers for glyphs is the task of ISO/IEC 10036, and AFII (Association for Font Information Interchange) is the current registration authority. To ensure global uniqueness, the ISO/IEC 10036 glyph identifiers are structured names as defined by ISO/IEC 9541.

The second aspect of the SC 18 definition of a glyph is that it is an abstraction that is independent of an actual image. This is analogous to the primary definition of a character as representing data. The level of abstraction is not defined; nor are criteria defined that would allow determining whether two potential images (forms) are instances of one abstract glyph, or are to be considered two distinct glyphs, each having an independent image.

The distinction between the concepts of *glyph* and *grapheme* is not addressed by this Technical Report. *Grapheme* is the concept used in linguistic theory in the following sense:[5)]

> **Allograph**: One of a group of variants of a grapheme or written sign. It usually refers to different shapes of letters and punctuation marks, e.g. lower case, capital, cursive, printed, strokes, etc., …
>
> **Grapheme**: A minimum distinctive unit of the writing system of a particular language, … the grapheme has no physical identity, but is an abstraction based on the different shapes of written signs and their distribution within a given system. These different variants, e.g. the cursive and printed shapes of letters M, m, cursivated m, M, etc. in an alphabetic writings system are all allographs of the grapheme /m/.

As can be seen, *glyph* and *grapheme* are clearly related, partly overlapping concepts. The difference is that the grapheme concept is defined in relation to writing systems of particular languages, whereas the glyph concept is defined independently of language.

C.2. Assignment of glyph identifiers

In specifying characters for inclusion in a character set standard, SC 2 normally has recourse to the meaning of a character, and, in particular, has the option of unifying two or more forms if it is determined that those forms do not represent distinctions in meaning within a particular written language, or that the forms represent merely stylistic differences. In registering glyphs, the glyph registration authority of ISO/IEC 10036 has recourse to analysis of the form of the glyph, and has worked to identify which potential glyphs are merely

[5)] R. R. K. Hartmann and F. C. Stork, *Dictionary of language and linguistics*.

design variations of a single abstract glyph. However, the glyph registration authority of ISO/IEC 10036 must be prepared to register an arbitrary glyph if so requested.

The difficulty of identifying design or writing system variants of a glyph is that the criteria for identifying distinct glyphs are culturally dependent. In Latin fonts used with European languages, a wide set of variations is allowed in the design of the glyphs. The skeletal structure of the glyphs can change; strokes can be omitted; the form of the stroke can change; and extra elements and some flourishes can be added without creating a new glyph. The users of ideographic glyphs are much more restrictive in the set of variations they will allow before a new glyph is created. Thus, the input of experts is extremely important in identifying the relevant glyphs to be registered.

C.3. Use of glyph identifiers

Glyph identifiers are typically used in the following data structures: (1) a font resource to uniquely identify the glyph metric and shape information contained in that font resource, (2) a character-to-glyph mapping table to identify the glyph(s) to be used when one or more character codes occur in a revisable document, (3) a glyph-index-map to identify the glyph to be used when a glyph index occurs in a formatted document, and (4) a glyph collection to identify the set of glyphs making up the collection. In these four uses, the industry is better served by having commonly-defined universal glyph-identifiers. However, fonts are not required to use registered glyph identifiers. For example within a font, ISO/IEC 9541 specifically allows the use of glyph identifiers that are not registered under ISO/IEC 10036.

C.3.1 Font resource

ISO/IEC 9541 defines a *font resource* as:

> *A collection of glyph representations together with descriptive and font metric information which are relevant to the collection of glyph representations as a whole.*

Each glyph representation in a font resource defines the metric and shape information associated with a specific glyph. It is necessary that each glyph representation be uniquely identified from all other glyph representations in that font resource. The glyph identifiers used within a font resource may be unique to that one font resource only, or may be unique within some larger scope (company register, industry register, national register, or international register).

C.3.2 Character-to-glyph mapping table

Character-to-glyph mapping tables are not defined by ISO standards, but are necessary to show the relationship between the character codes of a given coded character set standard and the glyph identifiers of a given font resource. A character-to-glyph mapping table is used in document formatting to identify which glyph identifier or identifiers should be used for presentation when a given character code or code sequence is encountered in a revisable document. For one-to-one mappings, the character-to-glyph mapping table is simplistic or non-existent. But, for many-to-one, one-to-many, or many-to-many mappings, the character-to-glyph mapping table may become quite complex and include metric information for repositioning component glyphs into composite shapes. The glyph identifiers used in a character-to-glyph mapping table may be the same as those used in the associated font resource, or may be indirectly mapped to the associated font resource.

C.3.3 Glyph-index map

Glyph-index maps are defined by ISO/IEC 10180 as a data structure that maps index values (presentation codes) in a formatted document to the glyph identifiers in an associated font resource. Such document formatting processes transform the character codes of an input document (using the information contained in a character-to-glyph map) into glyph-index numbers in a formatted output document. The formatting process will either dynamically build a glyph-index-map that uniquely associates the index values in the document to the glyph identifiers of the font resource, or it may use pre-defined (registered) glyph-index-maps.

C.3.4 Glyph collection

To aid in the process of identifying a font resource that contains a required set of glyphs, ISO/IEC 9541 defines a data structure called a glyph collection. A glyph collection data-structure is a list of glyph identifiers, and it may be assigned a unique identifier. Font resources may contain any combination of glyph identifiers, and revisable documents may contain any repertoire of character codes. In formatting and presenting a document, glyph collections help locate font resources that contain a full set of glyphs that correspond to the set of character codes contained in the document.

Annex D
Font models

D.1. Overview of font models

This annex describes three font models. The first two, the coded font model and the font resource model, are from SC 18. The third, the intelligent font model, is from the Unicode Consortium. Any one of these models could be used successfully to print or display characters coded in ISO/IEC 10646 or in other coded character sets. These font models rely not only on the processes described in this annex but also on the glyph data-structures described in Annex C.3, “Use of glyph identifiers”.

D.2. Coded font model

A *coded font* (or a *character-coded font*) is a data structure in which character codes are used to identify the glyph metric and glyph shape information contained in the font. In practice, two primary forms of this data structure are used; one in which the character codes are used directly in the font to identify the glyph metric and glyph shape information, and one in which the character codes are mapped to independent glyph identifiers contained in the font. The first form requires separate fonts for each code table supported, while the second form requires separate mapping tables for each code table supported (this later form saves storage). Both data structures depend on a one-to-one mapping of character codes to glyphs in a font, and this is the basis for the coded font model illustrated in Figure 7.

This font model is the historic presentation model for data processing. In this model, each character code encountered by the layout process is used to locate a corresponding glyph in the coded font. The glyph metric information for that character code is used to determine positioning of the glyph, along with line and page breaks. The formatted document may be interchanged to another location for presentation processing, or transmitted to a local presentation process. The presentation process would use the character codes contained in the formatted document to locate a corresponding glyph in the coded font, and use the associated

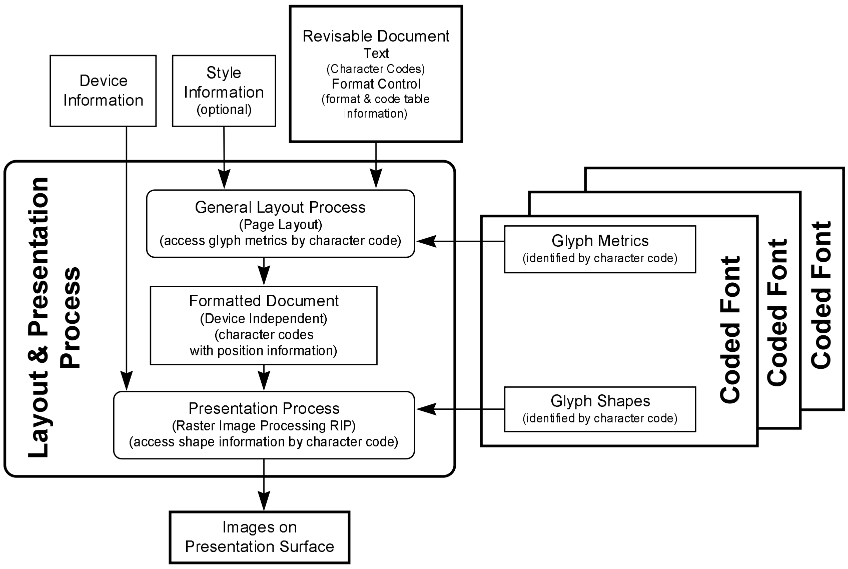
Revisable Document
Text
(Character Codes)
Format Control
(format & code table information)
Device Information
Style Information (optional)
Layout & Presentation Process
General Layout Process
(Page Layout)
(access glyph metrics by character code)
Formatted Document
(Device Independent)
(character codes with position information)
Presentation Process
(Raster Image Processing RIP)
(access shape information by character code)
Glyph Metrics
(identified by character code)
Glyph Shapes
(identified by character code)
Coded Font
Coded Font
Coded Font
Images on Presentation Surface

Figure 7 — Coded font model

glyph shape information to image the glyph on the presentation surface at the position indicated by the layout process.

Note that the coded font model is less suitable for the more complex glyph selection requirements of printing and publishing. For example, the Arabic script requires special processing in the coded-font model. If the input to the general layout process includes Arabic characters, the process also needs to convert the Arabic characters to the correct Arabic presentation forms.

D.3. Font resource model

The font resource model permits definition of font resources that are less dependent on any single coded character set or document-processing model. It is illustrated in Figure 8. This model is more suited to the document printing and publishing environment and permits blind interchange to occur between the layout and presentation processes. Glyph identifiers index the glyph metrics and glyph shape representations in the font resource. In this model, the layout process uses predefined character-to-glyph maps to determine the mapping (one-to-one, many-to-one, or one-to-many) of character codes to presentation glyphs and replaces the character codes in the formatted document with glyph index values. At the same time, the layout process builds a glyph index map (or it may use a predefined, registered glyph index map) that associates the glyph index values to the glyph identifiers used in the font resource.

The glyph index map is a mapping of glyph index values to glyph identifiers as shown in Figure 9 on the next page. The glyph index map may be

— unique to a particular indexed font,

— a mapping that is shared among several fonts, or

— a standardized mapping.

This flexibility allows a composition and layout process to generate a glyph index map that accesses only and exactly those glyphs of a large font resource that are needed to image the output of the process. This glyph index map may be combined with the font resource to produce an indexed font for this particular output.

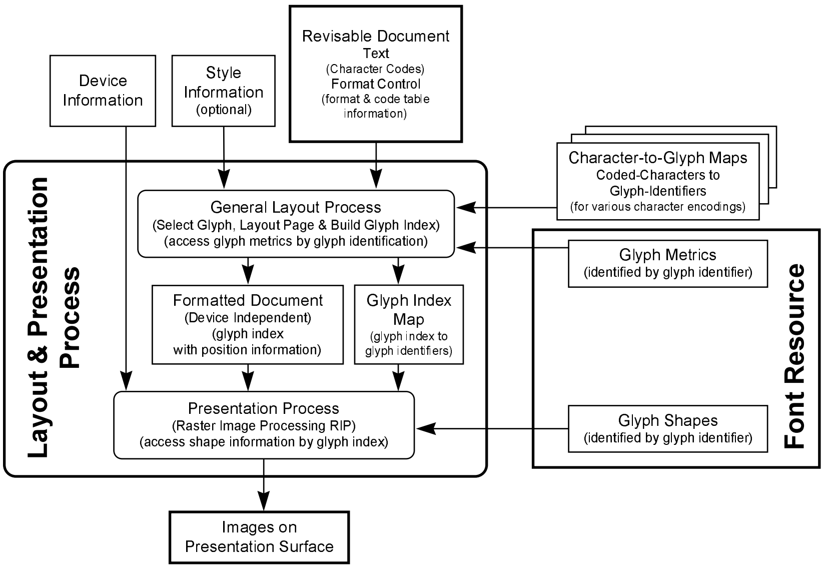
Revisable Document
Text
(Character Codes)
Format Control
(format & code table information)
Device Information
Style Information (optional)
Character-to-Glyph Maps
Coded-Characters to Glyph-Identifiers
(for various character encodings)
Layout & Presentation Process
General Layout Process
(Select Glyph, Layout Page & Build Glyph Index)
(access glyph metrics by glyph identification)
Glyph Metrics
(identified by glyph identifier)
Formatted Document
(Device Independent)
(glyph index
with position information)
Glyph Index Map
(glyph index to glyph identifiers)
Font Resource
Presentation Process
(Raster Image Processing RIP)
(access shape information by glyph index)
Glyph Shapes
(identified by glyph identifier)
Images on Presentation Surface

Figure 8 — Font resource model

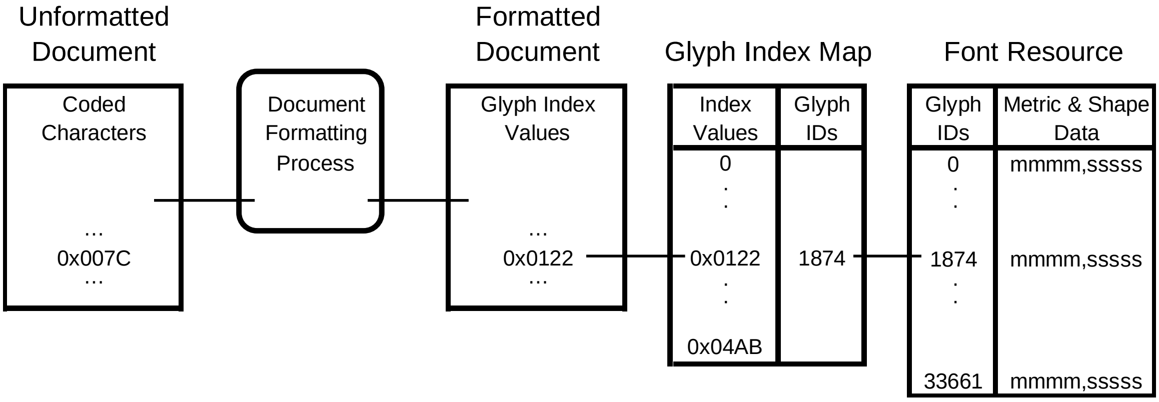

Figure 9 — Font resource, glyph index model

In the font resource model, the relationship between the character repertoire and the glyph collection may involve a one-to-one mapping, but may also involve a one-to-many or many-to-one mapping. It is essential for successful presentation that the set of glyphs in the glyph collection be mapable to the repertoire of characters used in the text or ideographic string. For the smaller, single-byte coded character sets, it is common to have a font resource that contains a glyph collection that contains all of the glyphs required to present the character repertoire of several coded character sets. However, for the larger ISO/IEC 10646 multi-octet coded character set, it will be more common to have font resources that contain glyph collections that are capable of presenting selected sub-repertoires of the total 10646 repertoire.

D.4. Intelligent font model

An intelligent font is a data structure that augments a font resource with additional information. The font resource contains:

— glyph representations

— glyph metrics

To this data structure, the intelligent fonts adds information describing:

— how a sequence of coded characters is transformed into a sequence of glyph identifiers, with associated position information

— how the transformation of coded characters to glyph identifiers is affected by style information

The first type of additional information typically includes several mappings from various coded character sets to private (font-specific) glyph identifiers. Subsequent transformations use the glyph identifiers. The subsequent transformations may be complex and may result in changes to the number and ordering of the glyph identifiers. For example, it may transform multiple coded characters into a single glyph (either because the glyph is a ligature or because the coded character sequence is a composite sequence), or a single coded character into multiple glyph representations that together construct the intended shape. See Annex E. The second type of additional information permits, for example, substitution of glyph subsets (e.g. swash variants, vertical substitution) based on style information.

An intelligent font can be used with a layout and presentation process that directly presents coded characters, that is, *plain text* (a coded character sequence that does not contain additional formatting information). Figure 10 on the next page shows the intelligent font model and the following paragraphs describe this model.

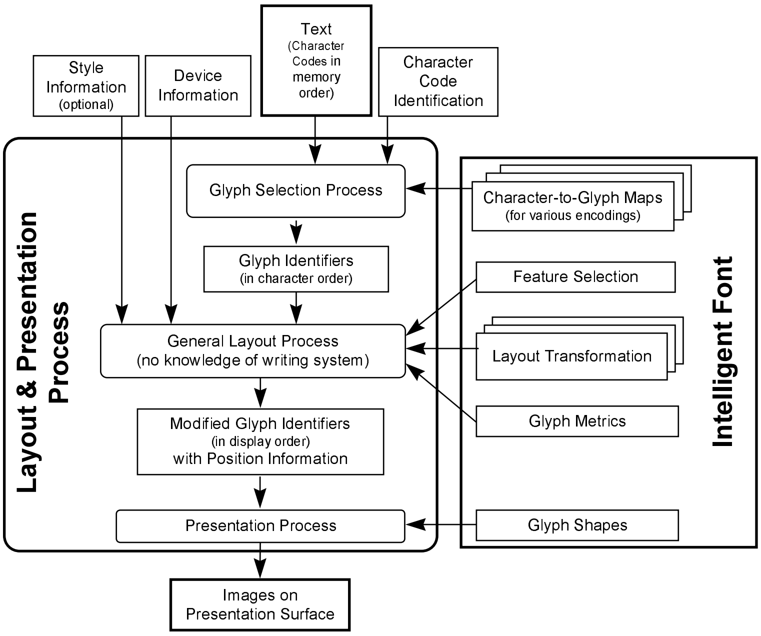
Style Information (optional)
Device Information
Text (Character Codes in memory order)
Character Code Identification
Layout & Presentation Process
Glyph Selection Process
Glyph Identifiers (in character order)
General Layout Process (no knowledge of writing system)
Modified Glyph Identifiers (in display order) with Position Information
Presentation Process
Images on Presentation Surface
Intelligent Font
Character-to-Glyph Maps (for various encodings)
Feature Selection
Layout Transformation
Glyph Metrics
Glyph Shapes

Figure 10 — Intelligent font layout and presentation model

Within the layout and presentation process of the intelligent font model, the glyph selection process transforms coded characters to glyph identifiers. This process requires:

— information about how the characters are coded

— the map from coded-characters to glyph identifiers for the specified character coding

The process takes coded characters in memory or logical order and produces glyph identifiers in character or logical order. Logical order is the order in which a person would normally read the characters regardless of the normal direction of the characters. Thus, for a text stream of Arabic which is written from right to left, the first character would be the rightmost character; for Latin which is written from left to right, the first character would be the leftmost character. For Latin text included in the middle of Arabic text, the logical order would be the rightmost Arabic character to the end of the Arabic text, then the leftmost Latin character to the end of the Latin text, and then the rightmost Arabic character of the second group of Arabic text to the end of the Arabic text.

Next the general layout process transforms the glyph identifiers in logical order into (possibly modified) glyph identifiers in display order. Display order is the order in which the characters are to appear on paper or on a screen. The general layout process requires:

— glyph metrics

— layout transformation

— feature selection information (how to use the optional style information)

— optional style information

— device information

The presentation process is the final process. It takes the glyph identifiers in display order, the glyph positions, and the glyph shapes to produce the images on paper or a screen.

D.4 Font Model Summary

Table 1 summarizes and compares the three font models described in this Annex. The primary difference between the three models is the sophistication of the glyph-selection process.

Table 1 — Comparison of Font Models

Characteristic		**Font Models**		
		Coded Font	**Font Resource**	**Intelligent Font**
Glyph Selection Process (character-to-glyph mapping)		None (1-to-1)	Yes (1 Process) (1-to-1 or M-to-N)	Yes (2 Processes) (1-to-1 or M-to-N)
Font Data-Structure	Character-to-Glyph Mapping	No (implied by character code position)	Yes (external to font resource)	Yes (in font resource)
	Index to Glyphs	Code Position in Code Table	Glyph Identifier (private or registered)	Glyph Identifier (private)
	Glyph Metrics and Shapes	Yes	Yes	Yes
	Additional Data	No	No	Feature Selection, Layout Transformation

Annex E
Examples of character to glyph mapping

E.1. Mapping characters to glyphs

This Annex shows examples of the character-to-glyph mapping process. It should be emphasized that it is often possible to represent a coded character sequence in more than one way and possible to provide a visual representation for it in more than one way. The two processes are separate, and they can be individually optimized.

E.2. One-to-one

A one-to-one mapping from character to glyph is the most frequently used in representing Latin-based languages, where the character U+0041 LATIN CAPITAL LETTER A “A”, for example, is likely to be drawn by using a single “A” glyph. The coded font model assumes that a one-to-one mapping is always the case.

It is often possible to use a single glyph to represent more than one distinct character. One example is U+00C5 LATIN CAPITAL LETTER A WITH RING ABOVE “Å” and U+212B ANGSTROM SIGN “Å” that both can be represented by the glyph “Å”. It is also conceivable for some implementations to use a single glyph for U+0041 LATIN CAPITAL LETTER A “A”, U+0391 GREEK CAPITAL LETTER ALPHA “Α”, and U+0410 CYRILLIC CAPITAL LETTER A “А”. These examples are different from the many-to-one mapping discussed below. Note also, that if a desired glyph is not coded, then the glyph cannot be used. For example, if the glyph for a fi ligature, “ﬁ”, is not in the coded font, the glyph is unavailable for display or printing.

E.3. Many-to-one

Many-to-one mappings are common even in Latin typography. The sequence U+0066 LATIN SMALL LETTER F “f” and U+0069 LATIN SMALL LETTER I “i” could be drawn by using a single glyph “ﬁ” for the ligature of “f” and “i”. The sequence U+0031 DIGIT ONE “1”, and U+2044 FRACTION SLASH “⁄”, and U+0032 DIGIT TWO “2” could be drawn by using a single “½” glyph.

Such mappings are more common in other writing systems. Hebrew, for example, makes extensive use of diacritical marks that are written around and even within various letters of the alphabet. The exact position of the diacritical marks varies depending on the letter with which they are written. The sequence U+05E4 HEBREW LETTER PE “פ”, and U+05BC HEBREW POINT DAGESH OR MAPIQ “◌ּ” and U+05B8 HEBREW POINT QAMATS “◌ָ” is often drawn by using a single glyph “פָּ” to provide optimal placement of the diacritical marks.

Level 3 implementations of ISO/IEC 10646-1 also use combining characters to represent accented Latin letters. Again, individual glyphs can be used to provide the best alignment of letter and accent. A level 3 implementation of ISO/IEC 10646-1 might well use the coded character sequence U+0065 LATIN SMALL LETTER E “e” and U+0301 COMBINING ACUTE ACCENT “◌́” but draw it using a single “é” glyph.

E.4. One-to-many

One-to-many mappings are more common than is often suspected. Whereas high-quality typography would insist on a large number of glyphs to provide greatest visual appeal, systems that cannot afford the necessary overhead can resort to other schemes. They might draw a U+00E9 LATIN SMALL LETTER E WITH ACUTE “é” by drawing the “e” glyph first then positioning the “´” glyph above it to form the glyph for the “é”.

One-to-many mappings are also found in Indic languages, where vowels can be written in two pieces, one on either side of the character they follow. The single character U+09CB BENGALI VOWEL SIGN O “◌ো” can be displayed using two glyphs that appear on either side of the related consonant.

ISO/IEC 10646 also included characters for Roman numerals. A system may choose to draw U+2165 ROMAN NUMERAL SIX “Ⅵ” by drawing a “V” and an “I” to the right.

E.5. Many-to-many

Given the previous examples, it should not be surprising that even many-to-many mappings occur. For example, in writing Vietnamese using level 3 of ISO/IEC 10646, the coded character sequence U+0065 LATIN SMALL LETTER E “e”, and U+0302 COMBINING CIRCUMFLEX ACCENT “◌̂” and U+0323 COMBINING DOT BELOW “◌̣” could occur. Displaying this sequence would require drawing an “e” with a “ˆ” above it and a dot “ ̣ ” below it. A system that has an “ê” glyph may choose to use that glyph and then add the dot below, and a system that has a single glyph for this sequence may simply draw that. (Similarly, a Level 1 implementation of ISO/IEC 10646 would use the coded character U+1EC7 LATIN SMALL LETTER E WITH CIRCUMFLEX AND DOT BELOW “ệ”.)

Indeed, depending on the details of the individual implementation, many of the examples from the previous clauses could be recast in a many-to-many fashion. Again, note carefully that depending on the individual designs of the glyphs, individual presentation systems will often differ in how they represent characters and how they present the associated glyphs.

Annex F
Recommendations of the original report

At its meeting held on 1-5 November, 1993, ISO/IEC JTC 1/SC 2/WG 2 (WG 2) received the original draft of the "Character-glyph model" (WG 2 document, N 915, dated 23 September, 1993). At the meeting, WG 2 resolved to accept the document as a first working draft of this Technical Report and requested a change to the "Purpose" clause (WG 2 document, N 949 R, dated 30 November, 1993). The requested change in purpose became item 4 in this list of recommendations.

1. In accordance with ISO/IEC 10036, AFII should undertake to register a comprehensive set of glyphs (graphic symbols) needed for each known writing system.

2. To facilitate the formatting and presentation of ISO/IEC 10646 coded character data, a set of associations between characters coded in 10646 and glyphs registered according to ISO/IEC 10036 should be defined. In particular, AFII should provide a table to document the ISO/IEC 10036 glyph identifier (or in the case of East Asian ideographs, the glyph identifiers) used to print each code position in the ISO/IEC 10646 standard.

    a. The term "association" in this context means that some glyph is suitable for presenting a character or a sequence of characters under appropriate circumstances.

    b. At least one glyph should be associated with each character.

    c. A character may be associated with multiple glyphs; likewise, a glyph may be associated with multiple characters.

    d. Some glyphs may not be associated with any single character; other glyphs may be associated only with a sequence (string) of characters.

3. The coding of additional presentation forms in ISO/IEC 10646 should be avoided. Rather, such forms should be registered as glyphs in accordance with ISO/IEC 10036.

4. The registration of additional glyphs in accordance with ISO/IEC 10036 should be avoided when

    a. the proposed glyph shares the same shape and associated glyph properties as a glyph already registered and

    b. the proposed glyph is distinguished solely by being associated with a different character.

5. SC 2 and SC 18 should adopt a common definition of terms and use the same terminology in developing standards. If SC 2 and SC 18 are unable to reach consensus on terminology, then when appropriate, SC 2 and SC 18 standards should cross-reference terms for the other subcommittee.