1.0 Unification of the Unicode Standard and ISO 10646

During 1991, members of the Unicode Consortium met over a period of several months with representatives from the International Organization for Standardization (ISO) to pursue a single international character encoding standard. Both bodies recognized that developing a single, universal character code would be beneficial. Meetings in October of 1991 finally resulted in mutually acceptable changes to both Unicode and the ISO Draft International Standard (DIS) 10646 which merged their combined repertoire into a single numerical character encoding. The new relationship between the Unicode standard and DIS 10646 is discussed in this chapter and the prospective changes to the Unicode Standard, Version 1 (Volume 1) are summarized.

A new DIS 10646 reflecting the result of this merger effort was distributed for international ballot in January 1992; final editorial changes are expected to be completed by the summer of 1992. At that time, the Unicode Consortium intends to publish a revised standard (Unicode version 1.1) to reflect the additional characters introduced from the DIS 10646 repertoire, and to incorporate minor editorial changes. The term "DIS 10646" refers to this balloted draft, which is known formally as ISO/IEC DIS 10646-1.2 of 26 December 1991.

The combined repertoire presented in DIS 10646 is a superset of the Unicode 1.0 repertoire. The Unicode Consortium does not intend to move or eliminate any of the codepoints that have been published in Volumes 1 and in Volume 2, as herein amended. In light of the cooperative spirit prevailing between the two organizations, the Unicode Consortium expects that DIS 10646-1.2 will pass as the International Standard substantially unchanged; the entire discussion in this chapter assumes that Unicode 1.0 may be implemented now as it stands. The majority of early implementations should not be affected by any final editorial changes that may be made in DIS 10646.

1.1 Structure of the 10646 Draft International Standard

DIS 10646 defines two alternative forms of encoding:

- A 32-bit encoding conceptually divided into some 32,000 "planes," each containing 65,000 characters
- A 16-bit encoding encompassing Plane Zero.

The 32-bit form is referred to as "UCS-4" (Universal Character Set containing four bytes) and the 16-bit form is referred to as "UCS-2"(Universal Character Set containing 2 bytes).

The DIS 10646 nomenclature refers to coded characters as multiples of octets and assumes that octets are serialized, while the Unicode nomenclature refers to coded characters as indivisible 16-bit entities. DIS 10646 (clause 6.3) defines the "big-endian" order as canonical when text is serialized as a stream of bytes; this view conforms with existing ISO double-byte character standards. In some contexts, such as transmission over existing serial channels, the difference between these two views is negligible; in other contexts, such as storage in machine registers or disk drives when 16-bit coded characters are treated in a machine's native word order, the difference may be important. (See also the discussions in Volume 1, Section 2.6 and Appendix B, Implementation Guidelines.)

The code numbers from 0 through 65,535 decimal (FFFF hexadecimal) can be represented by character codes of 16 bits. This range is also called the "Basic Multilingual Plane" or BMP of DIS 10646. The standard is arranged so that the most useful characters (that is, characters found in all major existing standards worldwide) are assigned in this range. DIS 10646 does not define any characters in other planes.

Merging the Unicode standard and DIS 10646 consisted of aligning the numerical values of identical characters and then filling in some groups of characters that were present in DIS 10646 but not in the Unicode standard; as a result the character code values of ISO DIS 10646 UCS-2 and Unicode version 1.1 will be made precisely the same. The specific adjustments made to the Unicode standard in order to achieve this goal are listed in Section 1.3. Since DIS 10646 does not at present encode any characters outside of the BMP, the result is that the character repertoires and encoding assignments of the Unicode standard and DIS 10646 will be identical. The character "A", LATIN CAPITAL LETTER A, for instance, has the unchanging numerical value 41 hexadecimal. This value may be extended by any quantity of leading zeros to serve in the context of the following fixed-length encoding standards:

Bits	*Standard*	*Binary*	*Hex*	*Dec*	*Char*
7	ASCII	1000001	41	65	A
8	ISO 8859-1	01000001	41	65	A
16	Unicode	00000000 01000001	41	65	A
	(= 10646 UCS-2)				
	(= 10646 BMP)				
32	10646 UCS-4	00000000 00000000			
		00000000 01000001	41	65	A

This design eliminates the problem of disparate code values in all systems that use any of the above-named standards.

1.2 Relationship Between the Unicode Standard and ISO DIS 10646

The goal of merging the Unicode standard and ISO DIS 10646 was to make character code assignments in the Unicode standard and ISO DIS 10646 UCS-2 (that is, the ISO 10646 BMP) identical. Programmers and system users should be able to treat the character code values from the Unicode standard, UCS-2, and BMP as identities, especially in the transmission of raw character data across system boundaries.

A character code standard, however, consists of more than a chart of code values. It contains many peripheral ingredients that give it coherence and make it implementable. (These additional elements do not create incompatibility between the Unicode standard and ISO DIS 10646. They are summarized here in order to clarify the relationship between the two standards.) Also necessary to a complete standard is a set of complete functional specifications, and substantial background material designed to help implementers better understand how the characters interact and, in general, how best to implement the standard. The Unicode Consortium plans to continue offering workshops and supplemental materials to help implementers understand and make best use of the Unicode standard.

The Unicode Standard as a Subclass of ISO 10646

ISO DIS 10646 provides mechanisms for specifying a number of implementation parameters, generating what may be termed various "profiles" or "subclasses" of the standard. While ISO 10646 contains no means of explicitly identifying or "declaring" Unicode values as such, the Unicode standard may be considered as encompassing the entire repertoire of 10646 and having the following profile values:

- Form UCS-2 (16-bit codes)
- Implementation level 2 (allows both combining marks and precomposed characters)

Few applications are expected to make use of all of the 30,000-plus characters defined on the DIS 10646 Basic Multilingual Plane. The conformance clauses of the two standards address this situation in very different ways. DIS 10646 provides a mechanism for specifying included subsets of the character repertoire, permitting implementations to ignore characters that are not included (see Informative Annex G of DIS 10646). A Unicode implementation requires a primary level of including all character codes, namely by requiring the ability to store and retransmit them undamaged. Thus the Unicode standard encompasses the entire 10646 Basic Multilingual Plane without requiring that any particular subset be implemented. (However, the Unicode Consortium acknowledges the possibility that it may denote specific subsets that must be included for compliance in the future.)

The Unicode standard does not provide mechanisms for identifying a stream of bytes as Unicode characters, although it does supply the *byte order mark* (U+FEFF) to indicate byte ordering. ISO

10646 also allows the use of U+FEFF as a "signature" in Informative Annex E to DIS 10646. Since UCS-2 is equivalent in repertoire and encoding to the Unicode standard, this optional "signature" convention for discerning between forms UCS-2 and UCS-4 is brought to the attention of Unicode implementers. The method is summarized in section 1.3.

Character Names

Unicode character names follow the ISO character naming guidelines (summarized in Informative Annex J of DIS 10646). In general, the Unicode naming convention follows the ISO names themselves, but with some differences that are largely editorial. For example:

DIS 10646 name	029A	LATIN SMALL LETTER CLOSED OPEN E
Unicode name	029A	LATIN SMALL LETTER CLOSED EPSILON

In the ISO framework, the unique character name is viewed as the major resource for both character semantics and cross-mapping among standards. In the framework of the Unicode standard, character semantics are indicated via alias names, usage annotations, character properties, and functional specifications as mentioned below, while cross-mappings among standards are provided in the form of explicit tables. The occasional disparities between Unicode names and 10646 names do not cause any mapping problems since the code numbers themselves are identical. Version 1.1 of the Unicode standard will use ISO 10646 names while retaining cross references to the Unicode version 1.0 names.

Character Functional Specifications

The core of a character code standard is basically a list of canonical numbers standing for characters, but in some cases the semantics or even identity of the character may be unclear. Certainly a character is not simply the representative glyph used to depict it in the standard. For this reason, the Unicode standard undertakes to supply as much information as possible to clarify the semantics of the characters it encodes.

At a level above the implementation of specific scripts, the Unicode standard and DIS 10646 differ in the precise terms of their conformance specifications. Any Unicode implementation will conform to ISO 10646, but because Unicode imposes additional constraints on character semantics and transmissability, not all implementations that are compliant with ISO 10646 will be compliant with the Unicode specification.

1.3 Specific Changes to Align Code Positions and Repertoire

In the interests of supporting the single numerical encoding provided by both the Unicode standard and ISO 10646, the Unicode Technical Committee accepted the following changes to Unicode

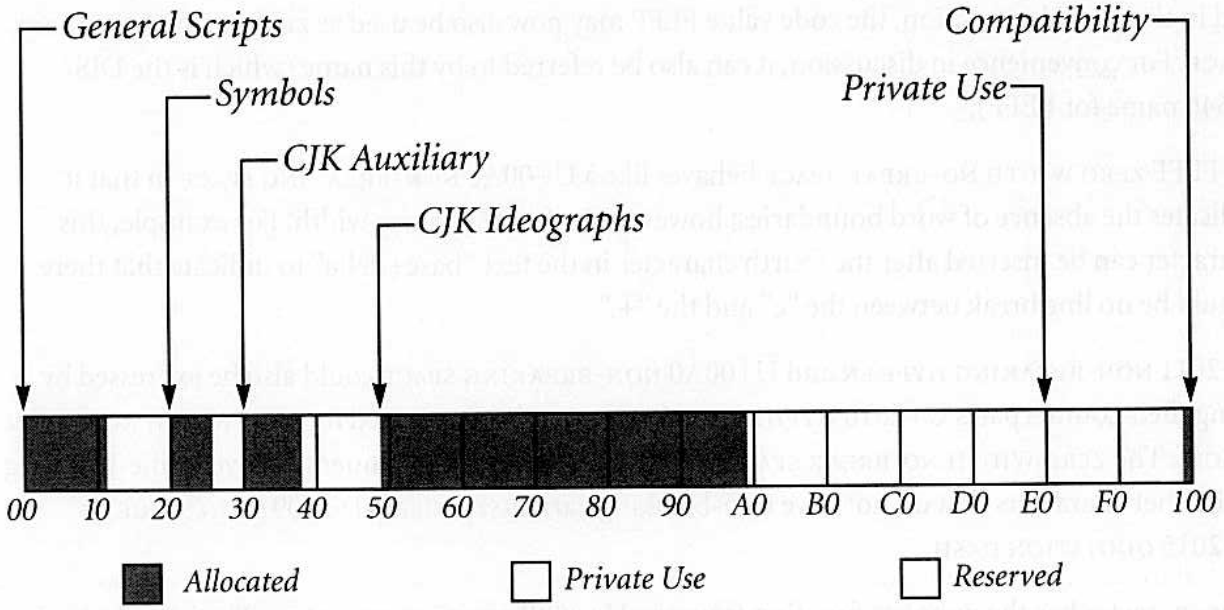
Version 1.0. (Readers may recall that the Notice in Volume 1 of *The Unicode Standard*, indicated that some changes were foreseen to consummate the merger with 10646.)

Block Changes

The following blocks have been either shifted or resized (note that there were no characters allocated here in Version 1.0 of the Unicode standard):

Block	*Old Range*	*Old Size*	*New Range*	*New Size*
Private Use Area	E800 → FDFF	5,632	E000 → F7FF	6,144
CJK Ideographs	4000 → 8BFF	19,456	4E00 → 9FFF	20,992
CJK Compatibility	(not coded)	720	F800 → FAFF	512

These changes make room for larger Korean Hangul sets and larger compatibility areas.



Characters Removed from the Unicode Standard

The following characters were removed from the Unicode standard in order to comply with ISO 10646:

U+04C5 CYRILLIC CAPITAL LETTER KA OGONEK
U+04C6 CYRILLIC SMALL LETTER KA OGONEK
U+04C9 CYRILLIC CAPITAL LETTER KHA OGONEK

U+04CA CYRILLIC SMALL LETTER KHA OGONEK

The preceding characters were unified with the following, respectively:

U+049A CYRILLIC CAPITAL LETTER KA WITH RIGHT DESCENDER
U+049B CYRILLIC SMALL LETTER KA WITH RIGHT DESCENDER
U+04B2 CYRILLIC CAPITAL LETTER KHA WITH RIGHT DESCENDER
U+04B3 CYRILLIC SMALL LETTER KHA WITH RIGHT DESCENDER

The following characters were removed entirely:

U+2300 APL COMPOSE
U+2301 APL OUT

The inclusion of these characters conflicted with further APL work.

Semantics of FEFF and ZERO WIDTH NO-BREAK SPACE

In addition to the meaning of byte order mark, as defined in Volume 1 of the Unicode standard and in the preceding section, the code value FEFF may now also be used as ZERO WIDTH NO-BREAK SPACE. For convenience in discussion, it can also be referred to by this name (which is the DIS 10646 name for FEFF).

U+FEFF ZERO WIDTH NO-BREAK SPACE behaves like a U+00A0 NON-BREAKING SPACE in that it indicates the absence of word boundaries; however, the former has no width. For example, this character can be inserted after the fourth character in the text "base+delta" to indicate that there should be no line break between the "e" and the "+."

U+2011 NON-BREAKING HYPHEN and U+00A0 NON-BREAKING SPACE could also be expressed by using their counterparts U+2010 HYPHEN and U+0020 SPACE bracketed by ZERO WIDTH NO-BREAK SPACES. The ZERO WIDTH NO-BREAK SPACE can also be used in this manner to prevent line-breaking with other characters that do not have non-breaking variants, such as U+2009 THIN SPACE or U+2015 QUOTATION DASH.

This character has the opposite function from the U+200B ZERO WIDTH SPACE. The latter indicates a word boundary, except that it has no width. For example, ZERO WIDTH NO-BREAK SPACE can be used to indicate word boundaries in scripts like Thai which do not use visible spaces to separate words.

The ZERO WIDTH NO-BREAK SPACE is not to be confused with U+200C ZERO WIDTH NON-JOINER. ZERO WIDTH NON-JOINER and U+200D ZERO WIDTH JOINER have no effect on word boundaries, while ZERO WIDTH NO-BREAK SPACE and ZERO WIDTH SPACE have no effect on joining or linking behavior. In other words, the ZERO WIDTH NO-BREAK SPACE and the ZERO WIDTH SPACE should be ignored when computing joining behavior; the ZERO WIDTH NON-JOINER and ZERO WIDTH JOINER should be ignored when computing word boundaries.

Further Semantics of FEFF

The code value FEFF is assigned a "signature" role in Informative Annex E to DIS 10646. Since UCS-2 is roughly equivalent to the Unicode encoding, this convention for discerning between forms UCS-2 and UCS-4 is recommended to the attention of implementers of the Unicode standard.

The code value FEFF, which in the Unicode encoding is *byte order mark* or ZERO WIDTH NO-BREAK SPACE, may be used at the beginning of a stream of coded characters to indicate that the characters following are encoded in the UCS-2 or UCS-4 representation, as follows:

Unicode encoding (or UCS-2) signature:	FEFF
UCS-4 signature:	0000 FEFF

Note that a character stream starting off with bytes FE and FF is unlikely to be ASCII text. Data streams (or files) that begin with 16-bit NULL followed by ZERO WIDTH NO-BREAK SPACE could be considered as likely to contain UCS-4 data; streams beginning with ZERO WIDTH NO-BREAK SPACE alone could be considered as likely to contain Unicode values. An application receiving data streams of coded characters may either use these signatures to identify the coded representation form, or may ignore them and treat FEFF as the ZERO WIDTH NO-BREAK SPACE character.

Additional Character Code Assignments

Assuming DIS 10646 is finalized in its current form, the Unicode standard will also annex approximately 5,000 additional elements from DIS 10646 into code positions that were unassigned in Unicode Version 1.0. As indicated in the Notice inVolume 1 of *The Unicode Standard* and mentioned previously, the Unicode Consortium intends to revise the Unicode standard late in 1992. Version 1.1 of the Unicode standard will be identical in character repertoire and code assignments to the approved International Standard ISO 10646.The principal groups of additional elements that will be added and published as part of the standard at that time are listed here. Page numbers refer to *ISO Draft International Standard 10646-1.2 of 26 December 1991.*

- Precomposed Latin diacritic combinations

  8 diacritic combinations, page 24

  3 digraphs, page 24

  24 diacritic combinations, page 26

  245 diacritic combinations, pages 74 and 76

- Precomposed Cyrillic diacritic combinations

  36 diacritic combinations, page 38

  6 Cyrillic clones of Latin characters, page 38

- Precomposed Greek diacritic combinations
  8 Greek double diacritics, page 78
  186 Greek diacritic combinations, pages 78 and 80

- Miscellaneous alphabetic presentation forms
  7 Latin ligatures, page 228
  6 Armenian ligatures, pages 40 and 228
  44 Hebrew presentation forms, page 228

- APL functional symbols
  9 CAD symbols, page 96
  68 APL functional symbols, page 96

- Arabic Koranic reading symbols
  24 Arabic Koranic reading symbols, page 46
  4 Arabic shaping controls, page 82

- Arabic ligatures
  176 Arabic positional forms, pages 228 and 230
  450 Arabic ligatures, pages 232, 234, 236, and 238
  2 Arabic punctuation presentation forms, page 236

- Korean Hangul combining elements
  93 Korean Hangul combining elements, page 118

- Modern Korean Hangul syllables from the standard KS C 5659-1990
  1930 modern Korean Hangul syllables

- Old Korean Hangul syllables from the standard KS C 5657-1991
  1754 old Korean Hangul syllables

- CJK time/date symbols

  1 telegraph symbol, page 112

  12 telegraph month symbols, page 122

  25 telegraph hour symbols, page 124

  31 telegraph day symbols, page 126