Appendix C

Relationship to ISO/ IEC 10646

Having recognized the benefits of developing a single universal character code standard, members of the Unicode Consortium worked with representatives from the International Organization for Standardization (ISO) during October of 1991 to pursue this goal. Meetings between the two bodies resulted in mutually acceptable changes to both Unicode Version 1.0 and the first ISO/IEC Draft International Standard DIS 10646.1, which merged their combined repertoire into a single numerical character encoding. This work culminated in the Unicode Standard, Version 1.1.

A second draft, DIS 10646.2, which reflected the result of this merger effort, was distributed for international ballot in January 1992 with a passing vote taken in late June 1992. After final editorial changes were made to accommodate the comments of voting members, the final standard was published in May 1993 as ISO/IEC 10646-1:1993, *Information technology—Universal Multiple-Octet*[1] *Coded Character Set (UCS)—Part 1: Architecture and Basic Multilingual Plane.* The Unicode Standard, Version 1.1 reflected the additional characters introduced from the DIS 10646.1 repetoire and incorporated minor editorial changes.

C.1 Timeline

Year	Version	Summary
1989	DP 10646	Draft proposal, independent of Unicode
1990	Unicode Pre-pub...	Pre-publication review draft
1990	Unicode 1.0	Edition published by Addison-Wesley
1990	DIS-1 10646	First draft, independent of Unicode
1992	Unicode 1.0.1	Modified for merger compatibility
1992	DIS-2 10646	Second draft, merged with Unicode
1993	IS 10646-1:1993	Merged standard
1993	Unicode 1.1	Revised to match IS 10646-1:1993
1995	10646 amendments	Korean realigned, plus 201 additions
1996	Unicode 2.0	Revised to cover 10646 amendments

The combined repertoire presented in ISO/IEC 10646 is a superset of the Unicode Standard, Version 1.0 repertoire as amended by the Unicode Standard, Version 1.0.1. The Unicode Standard, Version 1.0 was amended by the *Unicode 1.0.1 Addendum* in order to make the Unicode Standard a proper subset of ISO/IEC 10646. This entailed both moving and eliminating certain characters. The Unicode Standard, Version 2.0 covers the repertoire of

1. *Octet* is ISO/IEC terminology for *byte*, that is, an ordered sequence of 8 bits considered as a unit.

the Unicode Standard, Version 1.1 (and IS 10646), plus the first seven amendments to IS 10646, as follows:

1: UTF-16

2: UTF-8

3: Coding of C1 Controls

4: Removal of Annex G: UTF-1

5: Korean Hangul Character Collection

6: Tibetan Character Collection

7: 33 Additional Characters (Hebrew, Long S, Dong)

In addition, the Unicode Standard, Version 2.0 also covers Technical Corrigendum No. 1 (on renaming of Æ LIGATURE to LETTER), and such Editorial Corrigenda to ISO/IEC 10646 as were applicable to the Unicode Standard.

C.2 Structure of ISO/IEC 10646

ISO/IEC 10646 defines two alternative forms of encoding:

- A four-octet (31-bit) encoding containing 2^{31} code positions. These code positions are conceptually divided into 128 *groups* of 256 *planes,* each plane containing 256 *rows* of 256 *cells.*
- A two-octet encoding consisting of plane zero, the *Basic Multilingual Plane* (or BMP).

The 31-bit form is referred to as UCS-4 (Universal Character Set coded in 4 bytes) and the 16-bit form is referred to as UCS-2 (Universal Character Set coded in 2 bytes).

The code numbers from 0 through 65,535 decimal (0 - FFFF hexadecimal) can be represented by character code values of 16 bits. The most useful characters (that is, the characters found in major existing standards worldwide) are assigned in this range (that is, in the BMP). ISO/IEC 10646 does not currently define any characters in other planes.

Merging the Unicode Standard, Version 1.0 and DIS 10646.1 consisted of aligning the numerical values of identical characters and then filling in some groups of characters that were present in DIS 10646.1 but not in the Unicode Standard. As a result, the character code values of ISO/IEC 10646 UCS-2 and the Unicode Standard, Version1.1 are precisely the same. The Unicode Standard, Version 2.0 has added more characters, matching recent additions to ISO/IEC 10646-1:1993. The specific adjustments made to the Unicode Standard, Version 1.0 in order to achieve these goals are listed in *Appendix D, Cumulative Changes.*

Since ISO/IEC 10646 does not currently encode any characters outside of the BMP, the character repertoires and encoding assignments of the Unicode Standard and ISO/IEC 10646 are identical. For instance, the character "A", U+0041 LATIN CAPITAL LETTER A, 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 (see Table C-1).

This design eliminates the problem of disparate code values in all systems that use any of the standards just mentioned.

Table C-1. Zero Extending

Bits	*Standard*	*Binary*	*Hex*	*Dec*	*Char*
7	ASCII	1000001	41	65	A
8	8859-1	01000001	41	65	A
16	Unicode, UCS-2	00000000 01000001	41	65	A
31	UCS-4	0000000 00000000 00000000 01000001	41	65	A

C.3 UTF-16

The term UTF-16 stands for UCS Transformation Format for Planes of Group 00.

UTF-16 is the ISO/IEC encoding that is equivalent to the Unicode Standard with the use of surrogates as described in *Chapter 3, Conformance.* In UTF-16, each UCS-2 code value represents itself. Non-BMP code values of ISO/IEC 10646 in planes $1...16_{10}$ are represented using pairs of special codes. UTF-16 defines the transformation between the UCS-4 code positions in planes 1 to 16 of Group 00 and the pairs of special codes, and is precisely identical to the transformation defined in the Unicode Standard under D.28 in *Section 3.7, Surrogates.* Sample code for transforming UCS-4 into the Unicode Standard with surrogates is located in *Section A.2, UTF-8.*

As in the Unicode Standard, the first element of each surrogate code pair must be a UCS-2 code value in the range $D800\text{-}DBFF_{16}$; the second element must be a UCS-2 code value in the range $DC00...DFFF_{16}$. These two ranges are known as the *high-half zone* and *low-half zone,* respectively. Together, they constitute the newly defined S (Special) Zone of the BMP. Because each of these two ranges provide 1024_{10} values, a total of 1024^2 (= 1,048,576) code values may be represented through this mechanism. These code values are drawn from planes $1...16_{10}$ of group 0 of UCS-4, that is, the range of UCS-4 code values $00010000...0010FFFF_{16}$.

UTF-16 does not support the representation of all the UCS-4 code space but is limited to the BMP and the next 16 planes. This should not be an undue limitation since ISO JTC1/SC2/WG2 has stipulated that planes 1..14 will be filled first with future character assignments. Furthermore, of these additional planes, plane 15 ($000F0000...000FFFFF_{16}$) and plane 16 ($00100000...0010FFFF_{16}$) will be reserved for private use. There are other UCS-4 private use code values (in groups 60 to 7F and in planes E0 to FF in group 00) that are not accessible using UTF-16. Use of these private use code values is *strongly* discouraged because data encoded with these code values will not be interchangeable with Unicode implementations. Planes 15 and 16 should be used instead.

Applications interchanging ISO/IEC 10646 data containing non-BMP code values in planes 1..16 of ISO/IEC 10646 should use UTF-16 as the default encoding form in the absence of information to the contrary. Data exchanged with Unicode applications must be in UCS-2 form: if the data contains non-BMP encoded characters, they must be first transformed into UTF-16.

C.4 The Unicode Standard and ISO/IEC 10646

The goal of merging the Unicode Standard and DIS 10646.1 has been realized; making character code assignments *identical* in the Unicode Standard and ISO/IEC 10646 UCS-2 (that is, the ISO/IEC 10646 BMP). Programmers and system users should 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.

However, the Unicode Standard and ISO/IEC 10646 differ in the precise terms of their conformance specifications. Any Unicode implementation will conform to ISO/IEC 10646, Level 3, but because Unicode Standard imposes additional constraints on character semantics and transmittability, not all implementations that are compliant with ISO/IEC 10646 will be compliant with the Unicode Standard.

C.5 The Unicode Standard as a Profile of 10646

ISO/IEC 10646 provides mechanisms for specifying a number of implementation parameters, generating what may be termed various instantiations of the standard. ISO/IEC 10646 contains no means of explicitly declaring a profile matching the Unicode Standard as such. As a whole, however, the Unicode Standard may be considered as encompassing the entire repertoire of ISO/IEC 10646 and having the following profile values (as well as additional semantics):

- Numbered subset 300 (BMP)
- UTF-16 (if surrogates are used; UCS-2 otherwise)
- Implementation level 3 (allowing both combining marks and precomposed characters)
- Device type 1 (receiving device with full retransmission capability)

Few applications are expected to make use of all of the 38,000-plus characters defined in the ISO/IEC 10646 Basic Multilingual Plane. The conformance clauses of the two standards address this situation in very different ways. ISO/IEC 10646 provides a mechanism for specifying included subsets of the character repertoire, permitting implementations to ignore characters that are not included (see Informative Annex A of ISO/IEC 10646). A Unicode implementation requires a minimal level of handling all character codes, namely the ability to store and retransmit them undamaged. Thus the Unicode Standard encompasses the entire ISO/IEC 10646 Basic Multilingual Plane without requiring that any particular subset be implemented.

The Unicode Standard does not provide mechanisms for identifying a stream of bytes as Unicode characters, although to some extent this function is served by use of the *byte order mark* (U+FEFF) to indicate byte ordering. ISO/IEC 10646 also allows the use of U+FEFF as a "signature" as described in Informative Annex F to ISO/IEC 10646. Since UCS-2 is equivalent in repertoire and encoding to the Unicode Standard, Version 1.1, 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 the Specials subsection of *Section 6.8, Compatibility Area and Specials.*

C.6 Character Names

Unicode character names follow the ISO/IEC character naming guidelines (summarized in Informative Annex K of ISO/IEC 10646). In the prior version of the Unicode Standard, the naming convention followed the ISO/IEC naming convention,[1] but with some differences which were largely editorial. For example:

1. The names adopted by the Unicode Standard are from the English-language version of ISO/IEC 10646, even if other language versions are published by ISO.

ISO/IEC 10646 name	029A	LATIN SMALL LETTER CLOSED OPEN E
Unicode 1.0 name	029A	LATIN SMALL LETTER CLOSED EPSILON

In the ISO/IEC 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 in *Chapter 3, Conformance*, while cross-mappings among standards are provided in the form of explicit tables. The disparities between the Unicode Standard, Version 1.0 names and ISO/IEC 10646 names have been remedied by adoption of ISO/IEC 10646 names in the Unicode Standard. If the Unicode Standard, Version 1.0 name differed from the ISO/IEC 10646 name, then the previous name is provided as a cross reference in the *Unicode Character Database.*

C.7 Character Functional Specifications

The core of a character code standard is a mapping of code values to 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.

Thus the Unicode Standard consists of far more than a chart of code values. It contains many peripheral ingredients that give it coherence and make it implementable. Also necessary to a complete standard is a set of extensive character 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 Standard specifies properties and algorithms. Compliant implementations of the Unicode Standard will also be compliant with ISO/IEC 10646, Level 3; *however, not necessarily vice-versa.*