ISO
INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC 1/SC 2/WG 2

Universal Multiple-Octet Coded Character Set
(U C S)

ISO/IEC JTC1/SC2/WG2 N2176R
Date: 2000-03-06

Title: Implications of Normalization on Character Encoding
Source: Unicode Technical Committee
Status: Liaison
Action: For consideration by JTC1/SC2/WG2

As ISO/IEC 10646 / Unicode has become more prevalent in implementations and other standards, it has become necessary to produce very stable specifications for the comparison of text. In particular, a unique, normalized form of text is required for comparisons in domain names, XML element names, and other areas where a precise, stable, comparison of strings is required. Programs that require uniqueness also require forward compatibility: programs all over the web *must* be able to depend on the unique format not changing over time.

There are characters that are equivalently represented either as sequences of code points or as a single code point (called a *composite character*). For example, the *i* with 2 dots in *naïve* could be presented either as *i* + *diaeresis* (0069 0308) or as the composite character *i-diaeresis* (00EF). There are other cases where the order of two combining characters does not matter. For example, the pair of combining characters *acute* and *dot-below* can occur with either one first; both alternate orders are equivalent.

In response to the need for a unique form, the Unicode Consortium has produced an exact algorithmic specification of normalized forms. (For more information, see UTR #15: Unicode Normalization Forms.) One of these forms, Normalization Form C, is designed to favor precomposed characters such as ã over combining character sequences such as a + ~. The W3C *Character Model for the World Wide Web* requires the use of Normalization Form C for XML and related standards (this document is not yet final, but this requirement is not expected to change). See also the W3C *Requirements for String Identity Matching and String Indexing* for more background. We expect that the number of standards and implementations requiring normalization will continue to grow.

Such implementations must produce precisely the same result for normalization *even if* they upgrade to a new version of Unicode / 10646. Thus it is necessary to specify a fixed version for the composition process, called the *composition version.* The composition version is defined to be Version 3.0.0 of the Unicode Character Database, which corresponds to ISO 10646-1:2000.

To see what difference the composition version makes, suppose that Unicode 4.0 / 10646:2002 adds the composite *Q-caron*. For an implementation that uses Unicode 4.0 / 10646:2002, strings in Normalization Forms C or KC will continue to contain the sequence *Q + caron*, and **not** the new character *Q-caron*, since a canonical composition for *Q-caron* was not defined in the composition version.

The implications for encoding new characters are that new precomposed characters are important to recognize. If Q WITH CARON were added to Unicode 4.0 / 10646-1:2002, then it would represent a duplicate encoding. This could be tolerated before Unicode 3.0 because canonical equivalence could be used to equate the two forms. But due to the need for stability in comparison by so much of the world's infrastructure, this situation cannot be tolerated in the future. For stability, characters that are currently representable as sequences will always stay representable only as sequences. These include the following examples:

Character	Code Point Sequence	Comments
ch	0063 0068	Slovak, traditional Spanish
t^h	0074 02B0	Native American languages
	0078 0323	
	019B 0313	
	00E1 0328	LATIN SMALL LETTER A WITH OGONEK AND TILDE
	0069 0307 0301	LATIN SMALL LETTER I WITH DOT ABOVE AND ACUTE
	30C8 309A	Ainu in kana transcription

Moreover, the need for separate precomposed characters is diminishing quickly. The major GUI vendors are currently in the process of upgrading their systems to handle both surrogates and accurate positioning of combining marks, with such technologies as OpenType and AAT. By the time new precomposed characters could be added, there would be little need for them.

It *is* possible to add future precomposed characters in the case where they cannot already be represented by combining character sequences. In such cases the situation is reversed; the component characters that would make up an equivalent combining character sequence cannot be added.

References:

- OpenType: http://www.microsoft.com/typography/tt/tt.htm
- AAT: http://developer.apple.com/techpubs/macos8/TextIntlSvcs/ATSUI/ATSUI_ref/ATSUI-1.html
- UTR #15: http://www.unicode.org/unicode/reports/tr15
- Versions of the Unicode Standard: http://www.unicode.org/unicode/standard/versions/
- Unicode Character Database for Version 3.0: ftp://ftp.unicode.org/Public/3.0-Update/
- *Character Model for the World Wide Web*: http://www.w3.org/TR/WD-charmod
- W3C *Requirements for String Identity Matching and String Indexing* (http://www.w3.org/TR/WD-charreq)