ISO/IEC JTC1/SC2/WG2 N 3274

Date: **2007-06-18**

ISO/IEC JTC1/SC2/WG2
Coded Character Set
Secretariat: Japan (JISC)

Doc. Type: **Input to ISO/IEC 10646**

Title: **Proposal for a new edition of ISO/IEC 10646**

Source: **Project Editor**
Project: JTC1 02.18
Status: For review by WG2
Date: 2007-06-18
Distribution: WG2
Reference: WG2 N3229, N3230, N3275, N3276
Medium:

This document describes a proposal for a new edition of ISO/IEC 10646 and should be evaluated with the accompanying documents WG2 N3275 and N3276. It is based on preliminary considerations exposed in WG2 N3229.

With now five amendments being processed against the last edition of ISO/IEC 10646:2003, it becomes increasingly difficult to read the standard. This on itself would justify the creation of a new edition to reflect the consolidated content. In addition, the synchronization with the Unicode Standard presents a mix of challenges and opportunities. This document provides the rationale for the new edition.

1. Standard Architecture and Structure

1.1 Issues

1.1.1 Terminologies used by the two standards are not well aligned

ISO/IEC 10646:2003 uses the concept of UCS form which is sometimes interpreted as abstract notation (canonical form), memory representation, or serialized representation. Some of these forms use the 'UCS Transformation Format' moniker without clear description what it entails. There is no clear separation between the character coding space which is the space in which character code position/code point are created, and the code units in which characters are transmitted or stored. With the addition of UTF-8 and UTF-16, the current description has become increasingly confusing.

On another hand, the Unicode Standard has separated these concepts using code point mapped into a codespace to describe abstract characters, encoding forms for in-memory representation, and encoding scheme for serialization.

To describe the coding of characters (clause 6), ISO/IEC 10646:2003 uses a segmented view of characters into multi-octet sequences (G-P-R-C for Group-Plane-Row-Cell) which is unnecessarily complicated for what is in essence a 32-bit code unit. In addition with all the coding space beyond 10FFFF permanently reserved, it would be much easier to describe the coding space as a range extending from 0000 to 10FFFF.

In addition, serialized aspects of the coded representations are intermixed in various parts of the standard without clear separation between when a character coding needs to be serialized or not. If anything, amendment 3 with the introduction of new serialized encoding (UTF-32, UTF-32LE and UTF32-BE) makes the matter worse.

1.1.3 Essential part of the standards described in annexes

ISO/IEC 10646:2003 describes UTF-8 and UTF-16 in annex D and C respectively which may give the impression that these forms are not as important. In fact, UTF-8 is the preferred encoding forms for many applications and protocols in IETF and UTF-16 is widely implemented by operating systems. It would seem wiser to bring back these two UTFs into the main body of the standard.

1.1.4 Data set expressed through non machine readable list

Several key concepts of the standard (Combining characters and mirrored characters) are maintained by enumerated lists of characters that are not machine readable and are hard to maintain. For these concepts, the Unicode Standard simply maintains a set of properties available through machine readable files.

1.1.5 Lack of details in the name list

The ISO/IEC 10646:2003 in its clause 34 describes the character glyphs and names in a format that does not allow for extra comments and references short of a simple terse annotation. This creates the need for annex P which contains additional information about character. The solution used by the Unicode Standard is more flexible as it allows the same information and much more to be presented in the chart section.

1.1.6 Errors in character classes

Because characters were classified in enumerated lists without proper classification and maintenance, several errors have been introduced. This is especially the case for format characters. Many characters have been considered format characters which were not, such as 202F NARROW NO-BREAK SPACE, the Ideographic Description Characters, etc…At the same time some character which are format characters were not part of the list, such as 17B4 KHMER VOWEL INHERENT AQ, 2061 FUNCTION APPLICATION, all the TAG characters, etc…

1.2 Architecture updates

1.2.1 Terminology update

Introduction of the UCS code space which contains the coded characters represented in code point (same as code position). This can also be seen as the canonical representation of characters using a single integer between 000000 and 10FFFF for all coded characters. This also removes any reference to a coding space going beyond 10FFFF.

1.2.2 Introduction of the UCS encoding forms

This separates the canonical representation (UCS code points) from memory representation or interchange consideration. These encoding forms reflect the various encoding that can be used in memory to represent the code points (UTF-8, UTF-16, and UTF-32). UCS-2 is deprecated and UCS-4 is aliased with UTF-32. Along with this introduction, the CC-data-element concept is firmly associated with UCS encoding form data representation and the name is aliased with the ‘code unit sequence’ used by the Unicode standard.

This also allows moving the description of UTF-8 and UTF-16 from the annex section into the main normative body text.

1.2.3 Introduction of the UCS encoding schemes

This separates serialization considerations for the regular encoding and makes the explanation of the signature concept much clearer.

1.2.4 Introduction of character classification

Allowing the usage of Unicode General Category values makes the definition of all character types much simpler and guarantees synchronization with the Unicode Standard. In addition, the Bidi_Mirrored class replaces the enumerated list in annex B. In that aspect, table 1 in clause 6.3.1 is now a key element for character classification.

1.2.5 Format character overhaul

The format character concept is now aligned with the Unicode Standard (usage of GC: Cf, Zl, Zp). This has implied the move of TAG characters description into annex F, and the creation of a new annex (annex I) for the Ideographic Description Characters. Other non-format character descriptions formerly part of annex F have been moved to annex P.

1.2.6 New name list format

Although already introduced by amendment 5, the new version finalizes the switch to the new format by removing references to the '* 'annotation behind character names and aims at reducing significantly the size of annex P, once the current annotations are added in the Unicode name list.

1.2.7 Conformance clause update (clause 2)

The conformance clause incorporates the concept of the encoding forms and schemes. It also clarifies the processing of ill-formed CC-data-elements.

1.2.8 Combining characters (clause 20)

Replace the SPACE with NO-BREAK SPACE as the character to be used to create a composite sequence without the usual base characters. It also mentions that normalization does not eliminate multiple representations, but simply reduces the occurrence, mainly because 'multiple representations' is by itself a subjective matter. There are still many cases, especially with combining characters with combining class value of '0', where normalization would not re-order composite sequences that would be considered identical.

2. Document template

Since ISO/IEC 10646 was created, the ISO and IEC document templates (select 'Writing standards' in http://isotc.iso.org/isotcportal/index.html) have significantly changed, therefore creating a sizeable difference between current practice and the format used by ISO/IEC 10646. The main differences are:

- Multi-column layout versus single column. Current practice recommends usage of a single column layout which is more convenient for online access although the ISO/IEC 10646 'newspaper' format makes it slightly more readable when printed. However, single column layout inflates significantly the number of pages (20 to 30%). To minimize that size increase, many lists have been laid out in two columns, especially in Annex A (Collections).
- Better use of style sheets. The generation of the table of contents is now completely automated, including the annexes through the usage of adequate styles for headers.
- Separation of examples from main text part. Examples now use a smaller type and are indented to make clearer that they are not normative.
- Rationalization of list numbering styles. The usage of bullet, numbers, letters as list headers has been regularized to be better aligned with ISO/IEC standard writing style.

Other improvements, not part of the ISO/IEC templates have also been made:

- All cross-references are now active and can be accessed by clicking on them in PDF format.
- Usage of headers as bookmark in pdf has been improved, again by using appropriate header style in main text body and annexes.

3 Document status

3.1 Current presentation

The consolidated pdf document is provided in two formats:

1. Final document, with no revision markup (WG2 N3276)
2. Review document with revision markup (WG2 N3275)

To facilitate review, the reference document is not ISO/IEC 10646:2003, but instead a document where all five amendments have been incorporated, along with the template update and minor editorial corrections. The editor is well aware that three of these amendments are not finalized, but the intent is to synchronize the new edition with the updated amendments. This is only an issue with the two last amendments (4 and 5).

3.2 Open issues

3.2.1 Identification of features (clause 12)

Because there is no feature identification for the chosen encoding scheme, one has to be selected by default. Because historically, ISO/IEC 10646 has favored the big-endian mode (see clause 6.3 in ISO/IEC 10646:2003), it seems natural to automatically select the appropriate encoding scheme in the context of these ISO/IEC 2022 identifiers.

3.2.2 UTF-8 identification (clause 12)

Currently UTF-8 is identified by ESC 02/05 02/15 04/09 which implies through the usage of the 02/15 octet that the return to the restored state of ISO/IEC 2022 is not exactly as stated by ISO/IEC 2022 because of the possible padding in some encoding format (such as UTF-16 and UTF-32). But in the case of UTF-8, no padding is necessary, but still an identifier using 02/15 is used described normatively while the note in 12.2 (formerly in annex D.6) mentions another identifier not requiring 02/15, as in ESC 02/05 04/07, and therefore 'cleaner'. Why not just use that identifier?

3.2.3 Combining characters (clause 20)

The current text is still clearly lacking, especially clause 20.4 concerning multiple combining characters. To effectively address the issue, the concept of combining class needs to be added and a much larger portion of the Unicode Standard describing this aspect needs to be either added in ISO/IEC 10646 or at least referenced.

3.2.4 Annex A format

Annex A is still a hopeless mix of text sequences and normative number lists. These lists of numbers are nearly impossible to mine for use by application writers. It would be very useful to move most of the collection definitions into data files that could be easily parsed to reconstruct these collections. The result would be that the clauses A.3, A.4, A.5, and A.5 would mostly become simple introductions and format descriptions of a single file containing all these collections. However, clauses A.1 and A.2 would probably stay as they are.

3.2.5 Format characters (Annex F)

Annex F contains description for most but not all format characters and the format is not yet consistent. Furthermore they are not described in a very organized manner.

3.2.6 Empty Annexes

Many annexes are now empty. Is it worth maintaining a place holder, or simply deleting them? For referencing issues, it is probably wise to keep existing annex numbers (for example, annex L and annex S).

3.2.7 Annex P

The annex still contains many entries that should be removed by adding the annotations in the name list. This will require coordination with the editors of the Unicode Standard.
