

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

Doc. Type: Disposition of comments

Title: Disposition of comments on ISO/IEC CD 10646 5th edition

Source: Michel Suignard (project editor)

Project: JTC1.02.10646.00.00.00.06

Status: For review by WG2

Date: 2019-02-07

Distribution: WG2

Reference: SC2 N4635 N4646 N4416 WG2 N5013 N5014 N5016 N5018 IRG N2291

Medium: Paper, PDF file

Comments were received from China, Ireland, Japan, Korea (ROK), UK, and USA. The following document is the disposition of those comments. The disposition is organized per country.

Note – With some minor exceptions, the full content of the ballot comments have been included in this document to facilitate the reading. The dispositions are inserted in between these comments and are marked in **Underlined Bold Serif text**, with *explanatory text in italicized serif*.

Based on the number of negative ballots (5), and the number of conflicting demands, the editor has not tried to create dispositions that would recognize a consensus and allow a fast path to a draft amendment (DAM). Instead, the intent is to have a new CD ballot (CD.2) that will include all non-controversial technical and editorial comments, and many new non-controversial repertoire additions proposed since last WG2 that are supported by fully formed proposals for addition. Based on the disposition, it is likely that only one of the negative ballots could be reversed: Korea (ROK) because all the NB comments were accommodated.

Beyond the ballot comments, another expert contribution, WG2 N5018 written by Peter Constable, was considered in improving the 6th edition. The contribution is focusing on the terms and definition, and the clause 12 related to ISO/IEC identification. The editor felt that incorporating the results of these comments into the next CD version would be productive. As such, these suggestions are disposed in the last part of this document and create modest but useful changes to the text.

Concerning requests to remove repertoires, the dispositions were organized as follows:

- Remove repertoire that were requested to be removed by all parties. This concerns the Latin letters with overcurl: A7D0-A7D9.
- Preserve repertoires that some parties asked to be removed and some other parties asked to be preserved. This concerns CJK ideograph (Gonche) 9FF0-9FF6, Latin letters A7C0-A7C1, Zanabazar Square 11A48-11A49, Counting Rods 1D379-1D37D, Geometric shapes: 1F7E0-1F7EB, Symbols: 1F90D-1F90E, 1F979.

- Preserve the initial CJK Extension G repertoire and reorder (minor) it according to IRG #51. It is expected that IRG will keep working on feedback on CJK ext. G until their meeting #52. This also insures a reasonable stable reference (the reordered parts are extremely limited on scope) on which various IRG experts can keep building feedback.

There are two proposed addition as a direct consequence of these dispositions: 1DFA COMBINING OVERCURL and 1E94B ADLAM NASALIZATION MARK. Details about these changes are provided in the next pages. The other indirect additions (i.e. not related to these dispositions) are also briefly described in the next pages; details can be found in the referenced proposal documents.

In general, for the CD.2 ballot, NBs and liaison organizations are invited to repeat any ballot comments that were not accommodated to their satisfaction either by repeating their ballot text in the new ballot or referencing their comments indicated in these dispositions.

The following items reflect the repertoire changes to the CD between CD.1 and the planned CD.2

Characters added to CD as disposition of comments (supporting document # appended):

1DFA COMBINING OVERCURL (WG2 N4907)
1E94B ADLAM NASALIZATION MARK (WG2 N5022)

Characters removed from CD as disposition of comments:

A7D0 LATIN SMALL LETTER A WITH OVERCURL
A7D1 LATIN SMALL LETTER E WITH OVERCURL
A7D2 LATIN SMALL LETTER I WITH OVERCURL
A7D3 LATIN SMALL LETTER M WITH OVERCURL
A7D4 LATIN SMALL LETTER N WITH OVERCURL
A7D5 LATIN SMALL LETTER R WITH OVERCURL
A7D6 LATIN SMALL LETTER S WITH OVERCURL
A7D6 LATIN SMALL LETTER T WITH OVERCURL
A7D8 LATIN SMALL LETTER U WITH OVERCURL
A7D9 LATIN SMALL LETTER Y WITH OVERCURL

Characters for which removal was requested by some parties but stay (pending further comments in CD.2), one character moved: TROLL from 1F979 to 1F9CC

A7C0 LATIN CAPITAL LETTER THORN WITH DIAGONAL STROKE
A7C1 LATIN SMALL LETTER THORN WITH DIAGONAL STROKE

11A48 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER LA
11A49 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER SA

1D379 SOUTHERN SONG COUNTING ROD UNIT DIGIT FOUR
1D37A SOUTHERN SONG COUNTING ROD UNIT DIGIT FIVE
1D37B SOUTHERN SONG COUNTING ROD UNIT DIGIT NINE
1D37C SOUTHERN SONG COUNTING ROD TENS DIGIT FIVE
1D37D SOUTHERN SONG COUNTING ROD TENS DIGIT NINE

1F7E0 LARGE ORANGE CIRCLE
1F7E1 LARGE YELLOW CIRCLE
1F7E2 LARGE GREEN CIRCLE
1F7E3 LARGE PURPLE CIRCLE
1F7E4 LARGE BROWN CIRCLE
1F7E5 LARGE RED SQUARE
1F7E6 LARGE BLUE SQUARE
1F7E7 LARGE ORANGE SQUARE
1F7E8 LARGE YELLOW SQUARE
1F7E9 LARGE GREEN SQUARE
1F7EA LARGE PURPLE SQUARE

1F7EB LARGE BROWN SQUARE
1F90D WHITE HEART
1F90E BROWN HEART

1F9CC TROLL

30000..31349 CJK UNIFIED IDEOGRAPHS EXTENSION G

Repertoire reordering (minor) through this ballot

30000..31349 CJK UNIFIED IDEOGRAPHS EXTENSION G (see details in comment GE2 from Japan)

CJK Repertoire with K source updates (see details in ROK ballot disposition):

3EAC (delete K source), 8C6C (modified K source, same glyph), 248F2 (new K source, new glyph),
27CEF (new K source, new glyph), 2EB7E (modified K source, same glyph), 2EB89 (modified K source, same glyph)

Characters added to CD as UTC proposed additions, existing blocks (supporting document # appended):

0B55 ORIYA SIGN OVERLINE (WG2 N5023)

A82C SYLOTI NAGRI SIGN ALTERNATE HASANTA (WG2 N5024)

16FF0 VIETNAMESE ALTERNATE READING MARK CA (WG2 N4915, N5011, and N5026)

16FF1 VIETNAMESE ALTERNATE READING MARK NHAY

1F1AD MASK WORK SYMBOL (WG2 N5027)

1F8B0 ARROW POINTING UPWARDS THEN NORTH WEST (WG2 N5028)

1F8B1 ARROW POINTING RIGHTWARDS THEN CURVING SOUTH WEST

Characters added to CD as UTC proposed additions, new blocks (supporting document # appended):

Lisu Supplement 11FB0-11FBF (WG2 N5025)

11FB0 LISU LETTER YHA

Symbols for Legacy Computing 1FB00-1FBFF (WG2 N5028)

1FB00 BLOCK SEXTANT-1

...

1FB92 UPPER HALF INVERSE MEDIUM SHADE AND LOWER HALF BLOCK

1FB94 LEFT HALF INVERSE MEDIUM SHADE AND RIGHT HALF BLOCK

...

1FBCA WHITE UP-POINTING CHEVRON

1FBF0 SEGMENTED DIGIT ZERO

...

1FBF9 SEGMENTED DIGIT NINE

China: Positive with comments

Technical comment

T1. Mongolian

China does not comment on Mongolian but may submit proposals on the whole block later.

Proposed change by China


None.


Noted


Editorial comment

E1. Miao

China has comments on MIAO scripts:

16F2D  MIAO LETTER NYHA , U+16F2D is mainly used in Bai Yi. So, a sentence of “used in Bai Yi “ should be added to the note of this character.

16F2E  , delete “used for dzha in Dry Yi” of U+16F2E, for this character is mainly used in Miao.

16F32  , modify the note as “archaic character used before 1949 reformed orthography” of U+16F32. We have confirmed in the former discussion that this character was used before but not after 1949.

Proposed change by China

See above.

Accepted

Note that this editorial comment was also submitted to the DAM2 ballot to ISO/IEC10646 5th edition and is already accommodated in FDAM2.

Ireland: Negative

Ireland **disapproves** the draft with the technical and editorial comments given below. Acceptance of these comments and appropriate changes to the text will change our vote to approval.

Technical comments:

T1. Page 29, Row A72: Latin Extended-D

Ireland affirms its strong support for the encoding of A7C0 LATIN CAPITAL LETTER THORN WITH DIAGONAL STROKE and A7C1 LATIN SMALL LETTER THORN WITH DIAGONAL STROKE. Ireland notes that other Latin letters, such as 0244 Ƨ LATIN CAPITAL LETTER U WITH STROKE and 0289 Ƨ LATIN SMALL LETTER U BAR alongside A7B8 Ƨ LATIN CAPITAL LETTER U WITH DIAGONAL STROKE and A7B9 Ƨ LATIN SMALL LETTER U WITH DIAGONAL STROKE, are similarly distinguished for use in various languages—and *they were not encoded because they contrast in any single document*.

Capital and small Ƨ are used in a wide variety of languages, including Mesem and Melpa (languages from Papua New Guinea), Sayula Popoluca of Mexico, the Badwe'e language of Cameroon, the Budu language of Democratic Republic of Congo, Comanche, and Arhuaco of Colombia.

Capital and small Ƨ are used in the Mazahua language, an Oto-Manguean language recognized by statutory law in Mexico.

Opentype features are not used to distinguish Ƨ from Ƨ. Although not used in decomposition for these letters, 0335 COMBINING SHORT STROKE OVERLAY and 0337 COMBINING SHORT SOLIDUS OVERLAY are likewise distinguished in the UCS.

Capital and small Ƨ are used in Norse manuscripts and in the Nordicist typographic tradition for *þat*, *þes*, *þor*-. They form a pair which goes together with Ƨ, which is used for *þeim*, *þeir*. We do not have a great many examples of Nordicist texts using this character in print (one given in N4836R dates to 1846, and in N3027 examples date to 1828, 1846, and 1987). Nordicists are not unhappy with the encoding. We could not change the glyph for the former pair without disrupting Nordicist work.

Capital and small Ƨ are used in Old English and some Middle English manuscripts and particularly in the English typographic tradition dating back at least 452 years, for *þæt*. Document N4836R shows examples from 1566, 1623, 1644, 1659, 1665, 1689, 1705, 1709, 1714, 1715, 1737, 1828, 1845, 1875, 1882, 1889, 1909, 1914, 1930, 1956, 1959, 1960, 1967, 1973, 1991. This is a persistent scholastic convention which correctly represents the typical forms in Old English and Middle English manuscripts. Only in 2013 do we find the Nordicist glyph in use in transcriptions of Old English, because it was encoded on the basis of the 2006 proposal for medievalist additions to the UCS. (We also see it as a nonce character on one page only of an edition of *Beowulf* which was printed in 1882 and is likely to have been devised ad hoc by the printers.)

In 2006, when the first set of medievalist characters was being encoded, it was thought that that Anglicists would be satisfied with a generic thorn with stroke, but this is not the case. Some Anglicists have got used to the thorn with horizontal stroke, because it has been encoded, and using it is “better than nothing”. But it is not the culturally correct character for this tradition of scholarship, and it clashes with the culturally correct character for Nordicist use.

The Anglicist character is a high-frequency character in Old English and is also found in Middle English manuscripts and modern editions of texts in both languages. Both Nordicists and Anglicists are Germanicists, and no good is served to that discipline by an insistence that these characters must be unified. Nor could encoding these two character case any harm to the UCS or to implementors (many of whom ignore the A72 block entirely anyway). We are reminded of warnings of doom and confusion should Phoenician be disunified from Hebrew, or SINOLOGICAL DOT be encoded distinctly from MIDDLE DOT. No doom and no confusion resulted. Specialists can use those characters for their work. Anglicists deserve to be able to use the correct characters for theirs.

At the London meeting of WG2 in June 2018, in discussion it was mentioned that “in principle” an ascender or descender with a horizontal stroke could be considered to be “the same” as an ascender or descender with a diagonal stroke. All things being equal, this might be true. But the cultural conventions for Anglicists were persisted for *four and a half centuries*, and differ from those used by Nordicists. All things are not equal.

Opentype features should not be used to distinguish $\mathfrak{B}\mathfrak{p}$ from $\mathfrak{B}\mathfrak{p}$. An angled stroke in an \mathfrak{d} and an angled stroke in a \mathfrak{p} are what are expected in the English tradition. The UCS does not serve this expectation.

$\mathfrak{B} \neq \mathfrak{B}$
 $\mathfrak{p} \neq \mathfrak{p}$

Noted

See comment TE.31 from UK and TE.2 from US.

There is also a document WG2 N5013 (aka L2/18-286) which takes position on this issue.

T2. Page 1105, Row A72: Latin Extended-D

Ireland objects to the encoding of A7D0..A7D9 as atomic characters. We request their removal, and in their place request the encoding of a single COMBINING OVERCURL at 1DFA. Atomic encoding does not serve the user community for the COMBINING OVERCURL as described.



The *reason* this COMBINING OVERCURL has been proposed is for the purposes of palaeographic text representation and the study of encoded palaeographic text. Encoding a productive character in ten atomic precomposed characters is a bad idea for two reasons. First, it’s probably only a matter of time before the already-productive OVERCURL is found with some more Latin characters, like c or h or z, or with capital forms of the characters under ballot, and then those will have to be added as new atomic characters, involving the usual processing delays, which would be unnecessary if a single combining diacritical mark were available for use. Second, and more important, the OVERCURL is, truly, an abbreviation mark, and to be able to search and sort a letter with this abbreviation mark *as* an abbreviation mark, alongside and in comparison with other combining abbreviation marks, is an advantage for analysis. As atomic characters with no decomposition, the characters as they appear on the ballot would have no relation to one another. But in fact, they do. The letter *a* or *m* can bear 0306 ̊, 0311 ̋, 0352 ̌, or *1DFA ̍. Understanding the distribution and use of these marks is part of the *point* of palaeographic transcription.

We understand that an objection—or indeed the only objection—was a suggestion that font designers might not understand how to draw the characters. It is not clear why there should be such a suggestion—the overcurl is an end-stroke in the manuscripts, as shown in the proposal, so it would seem that for any Latin letter, a curl could be attached at any convenient place and then drawn in an arc over the letter. That’s what the scribes do. There aren’t really other choices. One should note that there appears to be no pre-existing *typographic* tradition with regard to the attachment of the OVERCURL. The mark has only recently been recognized and described as the abbreviation mark it is, and since palaeography and medieval Celticist and Anglicist use is relatively rare, no one to our knowledge has previously attempted to put it in type. (Of course, SC2 and the UTC have encoded many scripts and characters that have little or no typographic history.)

The proposal document N4907 indicates that even if a font doesn’t draw the OVERCURL with a pre-composed ligature, the unfused letter with OVERCURL is still legible. It notes another character, A7CD LATIN SMALL LETTER IS which also should fuse with its base letter but is legible if it does not. We note though that when we encode many scripts that are far more complex than Latin we do not require a full accounting of every ligature. Perhaps we should—but we have not done hitherto. In terms of the OVERCURL however, the requirement is simple. “The OVERCURL attaches at a convenient place at the end of the letter’s ductus, and swings over toward the left in an arc” is sufficient description of what is needed. Any competent typographer could manage

it. It is true that we have typically encoded Latin characters that merge with a diacritical mark as atomic characters. But these are usually used in natural language or phonetic orthographies. The COMBINING OVERCURL is explicitly an abbreviation character for specialized palaeographic use. As such, the right way to encode it is as a combining mark.

Recall please that very few fonts actually give high-level support the A72 block or the associated blocks of combining letters and other marks used for Medievalist or Germanicist use. Note too that other characters in the UCS, such as 0231..0232, and 1AB9..1ABA combine with base characters and no guidance regarding their fusion has ever been given. Note the frankly unusual typographic solution for 1D83, 1D8B, 1D8C.

We recommend that 1DFA replace the ten letters on the ballot. In addition, we recommend that the proposers of N4907 work to provide a Unicode Technical Note on the typography of the COMBINING OVERCURL, giving suggested forms for the whole Latin alphabet a..z and even A..Z if required—though so far only a subset of the lower-case Latin alphabet has been observed to use this mark.

Accepted

See also comment TE.32 from UK and TE3 from US.

Given that all NBs commenting on these characters are requesting the removal of the characters proposed at A7D0..A7D9, it makes sense to remove them from the CD.









The alternate proposal made in N4907 to encode 1DFA as follows will be incorporated in CD.2.

1DFA ◌ COMBINING OVERCURL







- used in medieval Cornish, English, Latin
- fuses typographically with at least a, e, i, m, n, r, t, u, y

T3. Page 1657, Row 1F78: Geometric Shapes Extended

Ireland requests that notice be taken of the request in N4913 “Towards dealing with hair styles and colouring in the UCS”. In this, and in N4011 “Proposal to add heraldic hatching characters to the UCS” it was pointed out that heraldic hatching, which is used in the black-and-white code chart glyphs in the UCS wherever colour has been specified, is partially represented already by a half dozen or so characters in the UCS. People interested in heraldry can make use of these, though some other colours were missing from the standard. The proposed emoji characters have blackand-white glyphs which will be suitable for heraldic use, but it would be better for the standard if those *existing* characters be given emoji status vis à vis colour, and that any additional characters required for emoji (and perhaps a few more for other heraldic purposes) be added to complete that set. Accordingly we believe that the following unifications and additions should be made:

 1F7E5 LARGE RED SQUARE	≡	 25A5 SQUARE WITH VERTICAL FILL
 1F7E6 LARGE BLUE SQUARE	≡	 25A4 SQUARE WITH HORIZONTAL FILL
 1F7E9 LARGE GREEN SQUARE	≡	 25A7 SQUARE WITH UPPER LEFT TO LOWER RIGHT FILL
 1F7EA LARGE PURPLE SQUARE	≡	 25A8 SQUARE WITH UPPER RIGHT TO LOWER LEFT FILL

and the following characters should be added to the standard:

 1F7E7 LARGE ORANGE SQUARE	→	 1F7E5 SQUARE WITH VERTICAL LINE AND DOT FILL
 1F7E8 LARGE YELLOW SQUARE	→	 1F7E6 SQUARE WITH DOTTED FILL
 1F7EB LARGE BROWN SQUARE	→	 1F7E7 SQUARE WITH VERTICAL AND UPPER LEFT TO LOWER RIGHT FILL

Note the glyph change for the character proposed as ORANGE. The hatching used for “orange” in the glyphs of some existing characters in the standard is the hatching for “tenné”, a light orange-brown, but if both brown and a *bright* orange are to be supported in the standard than a different hatching should be used for ORANGE and the few already-encoded glyphs using it should be altered. (We can supply new glyphs.) To support heraldry better, however, the following could also be added:

 1F7E8 SQUARE WITH VERTICAL AND UPPER LEFT TO LOWER RIGHT FILL

Supporting emojis is important, and we are glad to see colour extensions being proposed, but there is no reason the request to support heraldic tinctures cannot also be accommodated at this time. Four of the characters proposed on the ballot are duplicates of existing characters, and three, or preferably four, characters can be added to serve both emoji uses and heraldic ones.

(comment slightly edited to show actual size of proposed and already encoded characters)

Not accepted

See also comment TE.37 from UK.

There are several issues with the request:

- 1) The proposed characters are Emoji characters, the characters proposed for unification (U+25A5, U+25A4, U+25A7, and U+25A8) are not. There is a tendency among Emoji experts to avoid ‘emojification’ of existing characters.*
- 2) It is not clear that there is support to extend heraldic concept to new characters that are only targeting Emoji application.*
- 3) The relative sizes of the characters are different. The new characters are ‘large’ while the characters proposed for unification are medium.*
- 4) Unicode 12.0 content is frozen, and accepting this would break the synchronization.*

T4. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland affirms its support for the encoding of a new BILLIARD GAMES character at a suitable location, since (nearly) every vendor has changed the glyph of 1F3B1 BILLIARDS to that of the divination toy EIGHT BALL. The annotation given in the CD that BILLIARD GAMES “= magic 8-ball” violates character identity and the intent of the original emoji, and indeed causes confusion for anyone using a monochrome emoji font that follows the code charts. It makes no sense for all the vendors to be drawing this as an 8-ball but the code charts not to do so.



1F90C BILLIARD GAMES

→ 1F3B1  billiards

Noted

There is no broad consensus to add such character. Ireland can submit a proposal to add such character and make a similar comment in CD.2 ballot.

T5. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland affirms its strong support for the encoding of 1F979 TROLL in order to complete the set of fantasy beings. It would be inappropriate to unify the European troll (also used metaphorically for “internet trolls”) with the 1F479 JAPANESE OGRE (*oni* or *namahage*) and 1F47A JAPANESE GOBLIN (*tengu*) which do not have the common metaphorical use, and whose semantics are *explicitly* defined by the word JAPANESE.



1F979 TROLL

The Irish NB has not proposed this character as an emoji, but to complete the set of fantasy beings, though in the light of today’s political and social media environments, it is difficult to think that a character for TROLL wouldn’t find a great deal of popular use. To the complaint that “ogres” and “giants” might look like trolls, we say that the glyph provided could suit any of them, though the word “troll” has much more currency and the “giant” metaphor can be met by the MAMMOTH. To the complaint that encoding a symbol might encourage pressure on vendors to implement a troll, we say that it would firstly be a welcome character, and secondly that it is the vendors themselves who actively fuel the drive to add more and more characters for use as emojis. We oppose the removal of this character.

(code point for the proposed character corrected by editor)

Noted

See comment TE.4 from US.

The character will be balloted again in CD.2. But Ireland needs to make a separate proposal for this character as requested by the US. The location was changed to 1F9CC for type consistency within the block.

T6. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland affirms its request for the encoding of 1F9AB SQUIRREL. The attempt to unify squirrel with CHIPMUNK is an inappropriate over-extension of squirrel for the source character CHIPMUNK which was encoded for compatibility with the Windows webdings character set.



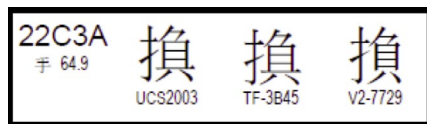
1F9AB SQUIRREL.

Noted

The case for SQUIRREL is weak as current vendor implementation do not differentiate between the two forms. There is no formal proposal for this character addition.

T7. Page 2671, Row 3050: CJK Unified Ideographs Extension G

Ireland requests the removal of U+3050C (UK-02790) from CJK Unified Ideographs Extension G as it is unifiable with U+22C3A. See attached image. (The T glyph form is anomalous, but as it has the reading *fù* it should be the same character as UK-02790.).



Not accepted

It is true that U+3050C 損 looks a lot like V2-7729. But the alternative solution may be to move V2-7779 to Ext G at the position U+3050C. This should be discussed by IRG experts at IRG #52 before deciding.

Editorial comments:

E1. Page 295, Row 20A: Currency Symbols

Ireland recommends that the glyph for 20BF BITCOIN SIGN be replaced with a more generic shape. It has always (since the advent of the EURO SIGN) been UCS practice to use glyphs for currency symbols which harmonize with the Times-style glyphs used for Row “000 C0 Controls and Basic Latin”. We can supply a suitable glyph.



Not accepted



















While it may be desired to harmonize with a Times-style glyph, the proposed picture may be a less optimal design in many aspects (placement of the bars, size of the bars, etc...). The current glyph is:



E2. Page 1407, Row 1343: Egyptian Hieroglyphs Format Controls

Ireland remains convinced that the glyphs for these characters must be revised. Control characters in the UCS typically have identifying letter abbreviations in their dotted-box glyphs. The exception to this is the Ideographic Description Characters, which are intended to be “visibly displayed graphic characters, not invisible composition controls” as the annotation to the code chart states. The glyphs of the Egyptian characters in this PDAM are hardly distinguishable from those of the Ideographic Description Characters, and we believe this










could be confusing to users and implementers. Discussion at the London meeting of WG2 in June 2018, the glyphs proposed in our comments on ISO/IEC 10646:2017 PDAM 2.3 were modified to increase the size and visibility of the symbols and to reduce the size of the abbreviation letters. We give these on the left below (on the right we give the previous glyphs for comparison).

	13430 EGYPTIAN HIEROGLYPH VERTICAL JOINER	
	13431 EGYPTIAN HIEROGLYPH HORIZONTAL JOINER	
	13432 EGYPTIAN HIEROGLYPH START AT TOP	
	13433 EGYPTIAN HIEROGLYPH START AT BOTTOM	
	13434 EGYPTIAN HIEROGLYPH END AT TOP	
	13435 EGYPTIAN HIEROGLYPH END AT BOTTOM	
	13436 EGYPTIAN HIEROGLYPH OVERLAY MIDDLE	
	13437 EGYPTIAN HIEROGLYPH BEGIN SEGMENT	
	13438 EGYPTIAN HIEROGLYPH END SEGMENT	













Not accepted

There is no requirement to have letter abbreviations in glyphs used for control characters, in fact many do not, and the only typical requirement is to use a dotted rectangular box. Therefore, there is no general principle in the UCS to have such abbreviations. In this case, the main information is to convey the type of layout adjustment related to a given control character. The acronyms bring very little practical information and also impose to make the embedded pictographs smaller. It is preferable to make these pictographs use as much of the content box as possible. However, the editor is open to improved glyphs.

For reference, the current glyphs are shown here:

13430		EGYPTIAN HIEROGLYPH VERTICAL JOINER
13431		EGYPTIAN HIEROGLYPH HORIZONTAL JOINER
13432		EGYPTIAN HIEROGLYPH START AT TOP
13433		EGYPTIAN HIEROGLYPH START AT BOTTOM
13434		EGYPTIAN HIEROGLYPH END AT TOP
13435		EGYPTIAN HIEROGLYPH END AT BOTTOM
13436		EGYPTIAN HIEROGLYPH OVERLAY MIDDLE
13437		EGYPTIAN HIEROGLYPH BEGIN SEGMENT
13438		EGYPTIAN HIEROGLYPH END SEGMENT

In addition, the glyphs for the newly proposed character have little to do with the Ideographic Description Character glyphs as shown below:

0		4		8	
	2FF0		2FF4		2FF8
1		5		9	
	2FF1		2FF5		2FF9
2		6		A	
	2FF2		2FF6		2FFA
3		7		B	
	2FF3		2FF7		2FFB

E3. Page 1621, Row 1F10: Enclosed Alphanumeric Supplement

Ireland requests that the rubrics and notes associated with 1F10D..1F10F and above 1F16D..1F16F read “Creative Commons”, not “Creative Common”.

Accepted

E4. Page 1657, Row 1F78: Geometric Shapes Extended

Ireland requests that the rubrics “Colored circles” and “Colored squares” use the Oxford spelling “Coloured”. The same comment applies at 1F90D.

Not accepted

In the current names list, there are 6 occurrences of ‘color’ in annotation, 1 occurrence in a name, and 2 occurrences of ‘colour’ in annotation. While the Oxford spelling has been in general preferred for names (but see above), annotations has using either convention (see for example the case of center and centre).

E5. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland recommends that the following glyphs be used for the reference glyphs (basically white outlined rather than black silhouettes), for consistency with other glyphs in the standard:



1F995 SAUROPOD



1F996 T-REX



1F998 KANGAROO



1F99B HIPPOPOTAMUS



1F99C PARROT (glyph to be improved)



1F99D RACCOON



1F99E LOBSTER (glyph to be simplified)

Accepted in principle

Based on receiving a font with the updated glyphs. Note that the same comment and similar disposition were respectively made and disposed in WG2 N4995 (pdam2.3 disposition of comment) and the editor did not receive a font.

E6. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland believes that the glyphs for 1F9B8 SUPERHERO and 1F9B9 SUPERVILLAIN should be replaced with more anthropomorphic figures. The glyphs below are based on public-domain heroes; specific attributes of costuming can be altered, but the use on the PDAM of “caped smiley faces” makes no sense, so we recommend caped isotype figures as shown below. Alternatively, the glyphs could be more like many of the

upper-torso-and-head glyphs that have been used. But the emoticon-style glyphs are not suitable. We can provide a font.



1F9B8 SUPERHERO



1F9B9 SUPERVILLAIN

Accepted in principle

The current glyphs could definitively be improved. However, it is not clear that the new glyphs are distinctive enough. Same comment and disposition were made in WG2 N4995.

E7. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland believes that the glyph for 1F9E7 RED GIFT ENVELOPE should have a closer vertical hatching as other red-coloured glyphs in the standard do. We can provide the font.



1F9E7 RED GIFT ENVELOPE

Accepted in principle

Based on receiving a font with the updated glyph. Same comment and disposition made in N4995.

E8. Page 1662, Row 1F90: Supplemental Symbols and Pictographs

Ireland believes that the glyphs for the hair colour swatches should have the same structure as the EMOJI MODIFIER FITZPATRICK TYPE characters which serve a similar function. At present these are essentially images of human scalps, for which there is no reasonable use scenario that isn't rather gruesome. The glyphs below have the same wiggly box that the Fitzpatrick modifiers do, as well as person images which harmonize with the reference glyphs of human pictographs. We can provide the font.



1F9B0 EMOJI COMPONENT RED HAIR



1F9B1 EMOJI COMPONENT CURLY HAIR



1F9B2 EMOJI COMPONENT BALD



1F9B3 EMOJI COMPONENT WHITE HAIR

Accepted in principle

These characters are not modifiers and should not have frames in their representative glyphs. However, the glyph can be improved, based on receiving a font with the updated glyphs. Same comment and disposition were made in N4995.

Japan: Negative

General, Technical, and Editorial comments (noted as GE, TE, or ED)

Technical comments

TE.1. Page 1, 2. Normative references.

In clause 2 “Normative references”, some documents even other than Unicode Technical Reports have the link to www.unicode.org site. It seems that this change has been made following the recommendation from ISO/IEC JTC 1/SC2/WG 2 meeting #67 (Recommendation M67.21.) However, this recommendation is not endorsed by ISO/IEC JTC 1/SC 2. Therefore, Japan requests SC 2 to deliberate this recommendation.

The concerns from Japan about proposed text is there is no guarantees that the documents referred are maintained consistently with this standard, because it may be updated by the organization not under control by this committee.

Proposed change by Japan.

Change this clause back to the one in the previous edition.

Also change the URL of documents other than Unicode Technical Reports throughout the documents back to the one starting from "http://standards.iso.org/iso-iec/10646/...."

Not accepted

As mentioned by Japanese NB, the change was done according to recommendation M67.21 which was adopted unanimously by WG2 (with Japan experts present). The lack of endorsement by SC2 looks like an oversight and as suggested by Japan, it should be deliberated at the next SC2 meeting.

About concerns with the maintenance consistency for standard, it should be noted that the Unicode standard has a much better maintenance story than ISO. Unicode standard versions have been maintained since version 1.1 published in 1993, and datafiles go back to 1995. In that aspect, Unicode has proven to be an excellent data repository. At the same time, typically ISO only maintains availability of the last version of a standard.

While this aspect can be discussed at the next SC2 plenary in June 2019, it seems advisable to maintain the current status.

Finally, it should be noted that unlike what is suggested in the ‘Propose change by Japan’, ISO/IEC 10646 has historically contained many URLs linked to documents other than Unicode Technical Report. Therefore acting on that requested change would be impossible, because many of these URLs point to resources not available under "http://standards.iso.org/iso-iec/10646/...."

GE2. Page 2655, 33.5 Code charts and lists of character names – CJK Extension G

CJK Unified Ideograph Extension-G (“CJK-G” in short) code chart has been updated before SC 2/WG 2 meeting #67 in London following the discussion by IRG meeting #50, then it is adopted in this ballot text.

However, Japan has the concerns with the following.

- Japanese experts reviewed updated CJK-G code chart and found that so many things were not correctly updated following the discussion record from IRG meeting #50. Note Japan expert sent this feedback to IRG convener and technical editor with copying WG2 project editor.
- IRG, then, reviewed above feedback at IRG meeting #51 and issued new report with discussion records. Japanese experts carefully checked this report and confused very much because it concluded that there were so many incomplete or wrong records in the report from IRG meeting #50. On the other hand, many changes has been made on the rules IRG decided before, such as the unification rules, the way of stroke counting and the choice of indexed radicals.

Japan is seriously concerned with such ad hoc operation by IRG. Therefore, Japan still thinks Ext G are not matured yet. Japan would like to suggest IRG re-considering its procedure before proposing new CJK Unified Ideographs Extensions.

By the way, during this CD ballot period, CJK-G was discussed in IRG meeting #51 held in October 22nd thru 26th. See the recommendation IRG M51.2 in SC2 N4641. This obviously violates the following in ISO/IEC Directives, Part 1.

JA.1.1 Discussion during ballot period

... Documents out for ballot at Committee Stage or any later stage shall not be subject to formal discussion at any working level of JTC 1 during the balloting period.

Proposed change by Japan.

Remove CJK-G from this CD

Not accepted

The opinion of Japanese IRG experts was apparently not shared by other IRG experts. It was felt that the result out of IRG meeting #50 was good enough to progress for ballot in the CD as reflected in recommendation M67.18 (unanimous) and endorsed in SC2 resolution M23.04 (although the adoption of M23.04 was not unanimous, nobody objected or abstained on the CJK Extension G part of the resolution).

The review at meeting IRG#51 was extremely limited, resulting only of code point re-ordering due to radical/stroke update. The bulk of CJK Extension G feedback is still due. While it is not clear whether the level of Ext G discussion during IRG #51 rose to the level of 'formal discussion' the editor will make sure the repertoire is not under ballot during the next IRG meeting (#52 in May 2019 Hong Kong, China).

Note also the text in the IRG#51 editorial report

(<http://appsrv.cse.cuhk.edu.hk/~irg/irg/irg51/IRGN2327WS2015EditorialReport.pdf>) as:

Most of the inconsistencies among IRG#50 discussion record (IRGN2291) and IRGN2308RWS2015v6 raised by Japan were confirmed. Their further process were also decided.

...

The following records are all about discussions on Japan's comments (inconsistencies among IRG#50 discussion record IRGN2291 and IRGN2308RWS2015v6). There are 6 stroke counts and 1 glyph to be changed, the corresponding conclusions are marked in red.

Based on this (extracted from the document linked above), the following changes will be made on the CD.2:

3010B 𪛗 15.12 GZ-3542301 SC goes to 13, code point moved to U+3010D

3041C 𪛘 57.9 GZ-1912202 SC goes to 8, code point moved to U+3041A

309CE 𪛚 112.9 GZ-1861301 SC goes to 8, code point moved to U+309C9

30A35 𪛜 115.10 GZ-3271501 SC goes to 11, code point moved to U+30A37

30DCA 𪛞 154.10 GZ-4412301 SC goes to 12, code point moved to U+30DD1

30E21 𪛠 157.9 GZ-1231301 glyph change to 𪛡, no R or SC change

30E37 𪛢 157.17 GZ-2171101 SC goes to 18, code point unchanged

Resulting code chart is in <http://appsrv.cse.cuhk.edu.hk/~irg/irg/irg51/IRGN2327Codechart.pdf>

IRG members have accepted to conduct further review of CJK Extension G and corresponding updates can be agreed upon at the next IRG meeting (#52) and incorporated into the 6th edition.

TE.3. Page 1076, 33.5 Code charts and lists of character names - Gongche” characters

These “Gongche” characters used for musical notation as the symbol have different characteristic and different usage from CJK unified ideographs. Further, the differences of shape from base characters cannot be considered in the framework of existing unification process. The rule that the character having tiny slash should not be unified with the base character is not consistent with the unification rule adopted so far. Therefore, these characters should not be encoded as CJK Unified Ideograph.

Proposed change by Japan.

Assign these characters into the code point on the block for script and symbols, not CJK ideographs

Not accepted

There have been several opinions on how to encode these characters, either as symbols, CJK characters on their own block, or integrated into an existing CJK block. The 2nd solution was proposed in WG2 recommendation M67.10. This proved unpractical for code chart production reason. After all, these characters have radical and stroke count like any other CJK ideograph, information which cannot be easily conveyed in a symbol block. After further feedback, it was considered easier to add them to the more convenient CJK addition area (i.e. end of the URO block starting at U+9FF0).

A further consideration is found in quoting WG2 N4967:

Many characters for Gongche Notation have the same appearances with Chinese ideographs exactly, so the vast majority of them could be used isomorphic ideographs to indicate. Hong Kong SARG once submitted two ideographs which are just used in the lyrics of Yueju Opera as UNCs in IRGN1405R. And UTC has submitted other two ideographs which are used in the lyrics of Kunqu Opera and traditional Gongche Notation to WS2017.

This means that there no intrinsic differences between ideographs used in Gonche Notation that are isomorphic or not. After all, many CJK Unified ideographs are also used in symbolic context (such as digits).

Based on this, it seems preferable to keep the proposed encoding as it is.

TE.4. Page 1849, 33.5 Code charts and lists of character names – U+2278B

The glyph on G column of U+2278B should not be changed, because proposed glyph cannot be considered to be unified with existing one.

Proposed change by Japan.

Do not change the glyph. In case this is error correction, add the information in “Annex P. Additional information on CJK Unified ideographs.”.

Not accepted

For reference, code chart extract in ISO/IEC 5th edition:

2278B 僣 僣
心 61.9
UCS2003 GHZ-10207.12

As proposed in CD 6th edition:

2278B 僣 僣
心 61.9
UCS2003 GHZ-10207.12

While an entry to Annex P may be desired, it does not prevent fixing what is clearly a glyph error for the original source. See WG2 N4988 for further details.

TE.5. 33.5 Code charts and lists of character names – New G sources

As proposed in ISO/IEC JTC 1/SC 2/WG 2 N4988, the source reference in G column of following CJK unified characters are changed.

- from GKX-0631.02 to GHZ-31665.01 for U+3CFD
- from GE-313D to GZFY-28665 for U+6FF9
- from GE-3952 to GLK-421274 for U+809E
- from GE-3D37 to GHZR-63304.09 for U+891D

- from GHZ-10761.12 to GHF-0229 for U+21D4C

The source reference is normative information and this change may cause the compatibility problem on the data transformation table with existing implementation, therefore this change should be avoided.

Proposed change by Japan.

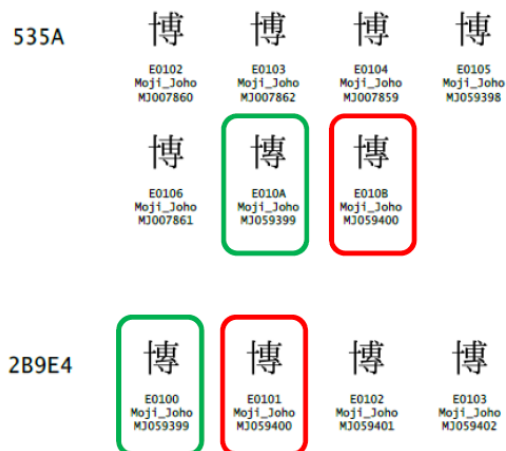
Do not change the source reference of these 5 glyphs on G column.

Not accepted

There are precedents to this kind of changes (including changes affecting J sources), and the requested was made by China (WG2 N4988) which should be well aware of possible impacts on data processing for its own sources. Source information typically determines the identity of encoded characters and may not necessarily be used as primary source for data transformation.

TE.6. Page 2768, A.5.10 390 MOJI-JOHO-KIBAN IDEOGRAPHS-2016 and A.5.11 391 MOJI-JOHO-KIBAN IDEOGRAPHS-2018

In IVD registration of Moji_Joho collection ([Registration of additional sequences in the Moji_Joho collection, dated 2017-12-12](#)), it is found that two different IVSs are registered for one glyph redundantly. As shown in the following figure, [535A E010A] and [2B9E4 E0100] are registered for Moji_Joho MJ059399, also [535A E010B] and [2B9E4 E0101] are registered for Moji_Joho MJ059400.



Proposed change by Japan.

- In JMJKI-2018.txt, delete <535A E010A> and <535A E010B> in the line starting from "535A."
- Associating with this change, add the following statement in Page 2768, Annex A.5.10 390 MOJI-JOHO-KIBAN IDEOGRAPHS-2016 as NOTE, such as "<535A,E010A> and <535A,E010B> have been removed in collection 391."

Accepted

ED.7. Page 34, Table 5 – GHR syntax

Proposed change by Japan.

Change "GHR-dddddd.dd" to "GHR-ddddd.dd."

Accepted in principle

The entry was in fact a duplicate of the previous entry (GHZR) and will be removed entirely along with its description in G sources enumeration.

ED.8. Page 48 Figure 8, Page 49 Fig. 9, Page 51 Fig. 10, Page 52 Fig. 11, Page 53 Fig. 12

Proposed change by Japan.

Change all "unabbreviated names" to "unabbreviated name" in NOTE under these figures.

Accepted

ED.9. Page 54, Figure 13

Proposed change by Japan.

Put the title of figure 13 on the same page of figure itself.

Accepted

ED.10. Page 2819, N.1 Methods of reference to character repertoires and their coding - First bullet in N.1

Throughout the document, this "international standard" has all been changed to "this document", but this one was left unchanged.

Proposed change by Japan.

Change "this International Standard" in first bullet to "this document".

Partially Accepted

See also comments made on the same subject by Peter Constable, especially E67.

While the wholesale replacement of "International Standard" is in general non-problematic, in this instance, the replacement is awkward, current text is:

- *Identification of this International Standard*

A reference to a character repertoire is related to the identification of the standard (in this case ISO/IEC 10646), not the document creating it. Replacing the above text by 'Identification of this documents' looks odd.

After consideration of comments made by Peter Constable on the same subject, it makes even more sense to revert to the original text as in:

- Identification of ISO/IEC 10646

Korea (ROK): Negative

Technical comment:

T1. Page 1997, CJK Unified Ideographs Extension B – U+248F2 and U+3EAC

K6-1002 is currently in U+3EAC. However, it seems proper to move K6-1002 to U+248F2.

If accepted by WG2, KR will send a TTF file containing a modified glyph of K6-1002.

See WG2 N5016 (= IRG N2349) for details.

Proposed change by Korea

Move K6-1002 currently in U+3EAC to U+248F2.

Accepted

The actual reference is K6-1022. The editor needs the font to process this change (another source used in the interim).

Note that document WG2 N5015 is asking to swap the K sources between 2EB7E and 2EB89 because it was an incorrect transcription from the original K submission. This will be accommodated in CD.2.

2EB7E	麥	2EB89	廢
麥 199.9		麻 200.4	
	KC-07185		KC-05976

Finally, the K source K1-6B6B was moved to U+27CEF with the glyph that was formerly at U+8C6C. U+8C6C has now a ghost K source KU-08C6C that will need to be changed or formalized.

E1. Page 33, 23.1 List of source references

Insert a blank between X and 1.

X1001, X1002, X1027-1, X1027-2, X1027-3, X1027-4, X1027-5.

Proposed change by Korea

X 1001, X 1002, X 1027-1, X 1027-2, X 1027-3, X 1027-4, X 1027-5.

Accepted

United Kingdom: Negative

General, Technical, and Editorial comments (noted as GE, TE, or ED)

ED.1. Page 12, 6.3.1 Classification (code points)

“The Table 1 summarizes the types”

“The” is unnecessary.

Proposed change by U.K.

Change to “Table 1 summarizes the types”.

Accepted

ED.2 page 24, 16.5 Ideographic description characters

“The Annex I describes them in more details”

“The” is unnecessary.

Proposed change by U.K.

Change to “Annex I describes them in more details”.

Accepted

ED.3. page 24, 16.6.1 General (Variation selectors and variation sequences)

“FVS1 To FVS3” should have lowercase “to”.

Proposed change by U.K.

Change to “FVS1 to FVS3”.

Accepted

TE.4. page 24, 16.6.1 General (Variation selectors and variation sequences)

“A variation sequence whose base character is a CJK Unified ideograph and whose variation selector is from the VARIATION SELECTORS SUPPLEMENT BLOCK is called an ideographic variation sequence.”

UTS #37 (since Revision 10) defines an ideographic variation sequence as being “a sequence of two coded characters, the first being a character with the Ideographic property that is not canonically nor compatibly decomposable, the second being a variation selector character in the range U+E0100 to U+E01EF”. This does not restrict an IVS base character to a CJK unified ideograph, but may be any character with the ideographic property (as long as it is not canonically or compatibly decomposable), including Tangut ideographs.

Proposed change by U.K.

Change to “A variation sequence whose base character is an ideograph which is not canonically or compatibly decomposable and whose variation selector is from the VARIATION SELECTORS SUPPLEMENT BLOCK is called an ideographic variation sequence.”

Add a note that at present the only ideographic variation sequences are for CJK unified ideographs.

Accepted

Though there is an issue with the term ‘ideograph’ which is not fully specified in ISO/IEC 10646. Doing some research, the editor could find some elements in the original proposal in

<https://unicode.org/L2/L2016/16291-ivd-proposal.pdf> but not in the current UTS #37 at

<http://www.unicode.org/reports/tr37/>. The original proposal had an enumeration of the proposed ideographs, UTS #37 specifies an ‘ideographic property’ but without formally establishing the value. Ideally, there should be a reference to an externally defined property (TBD) or a list of those code points.

The Note would read (previous note in sub-clause is renumbered NOTE 1):

NOTE 2 – At present the only specified ideographic variation sequences are for CJK unified ideographs.

TE.5. page 24, 16.6.2 Standardized variation sequences – add link to new Emoji file

“Standardized variation sequences are defined by a machine-readable format that is accessible as a link:
<https://www.unicode.org/wg2/iso10646/edition6/data/UCSVariants.txt>.”

The linked file does not define all standardized variation sequences, but refers to another file named “emoji-variation-sequences.txt” for “Emoji variation sequences”. However there is no link to this file, and it is not in the same directory, so it is not easy for a reader to find this file.

Proposed change by U.K.

Include the emoji-variation-sequences.txt file in the ISO/IEC 10646 data directory, and also include a link to that file in the appropriate subclause.

Accepted

TE.6. page 24, 16.6.2 Standardized variation sequences – restructuring of clause

“The Standardized Variations Sequences contain sequences associated with allowed base characters from the following categories:

Pictographic symbols.”

UCSVariants.txt (=StandardizedVariants-12.0.0.txt) refers to “Emoji variation sequences” which are listed in a separate file (emoji-variation-sequences.txt). Given that the Unicode Standard now defines variation sequences in three different places, as “Standardized variation sequences”, “Emoji variation sequences” and “Ideographic variation sequences”, it would be best for ISO/IEC 10646 to also refer to the three types of variation sequence separately.

Proposed change by U.K.

Remove mention of variation sequences for pictographic symbols from 16.6.2.

Add a new subclause 16.6.3 for Emoji variation sequences, which references emoji-variation-sequences.txt in the data folder for this standard.

Renumber 16.6.3 “Ideographic variation sequences” as 16.6.4.

Update 16.6.1 “General” to reflect the three types of variation sequence defined in the standard.

Accepted

TE.7. page 25, 16.6.3 Ideographic variation sequences – ideograph

“Variations sequences composed of a unified ideograph as the base character ...”

IVS base character is not restricted to a CJK unified ideograph, but may be any character with the ideographic property (as long as it is not canonically or compatibly decomposable), including Tangut ideographs.

Proposed change by U.K.

Change to “Variations sequences composed of an ideograph as the base character ...”.

Accepted

See disposition of comment TE.4 about the definition of ‘ideograph’.

TE.8. page 25, 16.6.3 Ideographic variation sequences – ideographic

“The purpose of IVD registration is to provide a technical solution to represent ideographs variants that are considered unifiable (see Annex S) and should not be encoded in CJK Unified ideographs.”

This only refers to CJK unified ideographs, but IVD registration is open to other ideographic scripts such as Tangut.

Proposed change by U.K.

Change to “The purpose of IVD registration is to provide a technical solution to represent ideographic variants that are considered unifiable (see Annex S for the unification rules for CJK Unified ideographs).”

Note change of ungrammatical “ideographs variants” to “ideographic variants”.

Accepted

See disposition of comment TE.4 about the definition of ‘ideograph’.

ED.9. page 34, 23.2 Source references file for CJK ideographs

“The Table 5 provides the format details”

"The" is unnecessary.

Proposed change by U.K.

Change to "Table 5 provides the format details".

Accepted

ED.10. page 36, 23.3.2 Source reference presentation for CJK UNIFIED IDEOGRAPHS block

"The figure 2"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 2".

Accepted

ED.11. page 37, 23.3.3 Source reference presentation for CJK UNIFIED IDEOGRAPHS EXTENSION A

"The figure 3"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 3".

Accepted

ED.12. page 38, 23.3.4 Source reference presentation for CJK UNIFIED IDEOGRAPHS EXTENSION B

"The figure 4"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 4".

Accepted

ED.13. page 38, 23.3.5 Source reference presentation for CJK UNIFIED IDEOGRAPHS EXTENSION C, D, E, F, and G

"The figure 5"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 5".

Accepted

ED.14. page 38, 23.4 Source reference presentation for CJK Compatibility ideographs

"The figure 6"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 6".

Accepted

ED.15. page 40, 24.2 Source reference file for Tangut ideographs

"The Table 6 pro-vides the format details"

"The" is unnecessary and hyphen should be removed.

Proposed change by U.K.

Change to "Table 6 provides the format details".

Accepted

ED.16. page 41, 25.2 Source reference file for Nüshu ideographs

"The Table 7 pro-vides the format details"

“The” is unnecessary and hyphen should be removed.

Proposed change by U.K.

Change to “Table 7 provides the format details”.

Accepted

ED.17. page 39, 24.2 Source reference file for Tangut ideographs

TangutSrc.txt states “The first is the UCS code point value as U+[x]xxxx (that is, there are either four or five hex digits)”, whereas in fact all Tangut ideographs and components are in the SIP so comprise five hex digits.

Compare NushuSrc.txt which states “The first is the UCS code point value as U+xxxxx (that is, there are five hex digits)”.

Proposed change by U.K.

Change comment at top of TangutSrc.txt to say “The first is the UCS code point value as U+xxxxx (that is, there are five hex digits)”.

Accepted

ED.18. page 39, 24.2 Source reference file for Tangut ideographs

TangutSrc.txt refers to “ISO/IEC 10646:2018 6th edition” on the first line and “ISO/IEC 10646:2017” on the third line..

Proposed change by U.K.

Remove “ISO/IEC 10646:2017” on the third line.

Accepted

ED.19. page 39-40, 24.2 and 25.2 – date

The dates in TangutSrc.txt and NushuSrc.txt do not seem to have been updated to reflect the latest edit date. The date should always be updated whenever the file is modified, even if it is just to update the version of the standard on the first line.

Also, the date format is different for the two files. NushuSrc.txt only gives the date, but TangutSrc.txt gives the day, date and time. The date format should be the same for all files attached the standard. We suggest that only the date is essential, and so day and time fields could be omitted.

Proposed change by U.K.

Ensure that all files attached to the standard (not just TangutSrc.txt and NushuSrc.txt) have the same format date stamp, and that the date stamp is updated whenever a file is modified.

Accepted

ED.20. page 39-40, 24.2 and 25.2 – file encoding

The description of TangutSrc.txt and NushuSrc.txt states: “The content linked to is a plain text file, using ISO/IEC 646-IRV characters with CARRIAGE RETURN/ LINE FEED as end of line mark.” However, inside each file is the comment that the value of the third field is encoded as UTF-8, which potentially contradicts the statement that the file uses ISO/IEC 646-IRV characters. In our opinion, it would be best to simply specify that the files are encoded as UTF-8, and not mention ISO/IEC 646-IRV at all.

This comment applies to the description of all files attached to the standard.

Proposed change by U.K.

For all attached files, change the description of the file encoding to specify “The content linked to is a plain text file, encoded as UTF-8 with CARRIAGE RETURN / LINE FEED as end of line mark.”

Inside each attached file, add a line indicating that the file’s encoding is UTF-8, and remove any specific mention of UTF-8 for a particular field.

Accepted

ED.21. page 40, 24.3 Source reference presentation for Tanguts ideographs

“The figure 7”

“The” is unnecessary and “figure” should be capitalized.

Proposed change by U.K.

Change to "Figure 7".

Accepted

ED.22. page 41, 25.1 List of source references (Nüshu)

No publication date is given for "Nüshu yongzi bijiao", which was published in 2006.

Proposed change by U.K.

Change to "Nüshu yongzi bijiao (2006)".

Accepted

ED.23. page 45, 26.10 Character names for Hangul syllables – Annex R

"The Annex R provides the names and annotation of Hangul syllables through a linked file."

"The" is unnecessary

Proposed change by U.K.

Change to "Annex R provides the names and annotation of Hangul syllables through a linked file."

Accepted

ED.24. page 45, 26.10 Character names for Hangul syllables – e

e) "Obtain the Latin character strings that correspond to the three indices I, P, F from columns 2, 3, and 4 respectively of table 8 below"

"table 8" should be capitalized.

Proposed change by U.K.

Change to "Obtain the Latin character strings that correspond to the three indices I, P, F from columns 2, 3, and 4 respectively of Table 8 below"

Accepted

ED.25. page 45, 26.10 Character names for Hangul syllables – h

h) "Obtain the Latin character strings that correspond to the three indices I, P, F from columns 5, 6, and 7 respectively of Table 6 below"

"Table 6" should be "Table 8".

Proposed change by U.K.

Change to "Obtain the Latin character strings that correspond to the three indices I, P, F from columns 5, 6, and 7 respectively of Table 8 below"

Accepted

ED.26. page 53, 30 Structure of the Supplementary Ideograph Plane (SIP)

"The figure 12"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 12".

Accepted

ED.27. page 53, 31 Structure of the Tertiary Ideograph Plane (TIP)

"The figure 13"

"The" is unnecessary and "figure" should be capitalized.

Proposed change by U.K.

Change to "Figure 13".

Accepted

ED.28. page 54, 32 Structure of the Supplementary Special-purpose Plane (SSP)

“The figure 13”

“The” is unnecessary and “figure” should be capitalized.

It should be Figure 14.

Proposed change by U.K.

Change to “Figure 14”.

Accepted

ED.29. page 54, 33.1 General (Code Charts and lists of character names)

“Detailed code charts and lists of character names for the BMP, SMP, SIP and SSP are shown on the following pages.”

TIP is missing.

Proposed change by U.K.

Change to “Detailed code charts and lists of character names for the BMP, SMP, SIP, TIP and SSP are shown on the following pages.”

Accepted

ED.30. 33.5 Code Charts and lists of character names – 4E00-9FFF

9FD4. Source glyphs are on two rows.

Proposed change by U.K.

Keep two source glyphs on a single row.

Not accepted

This is a limit of the code chart production tool. For the CJK Unified Ideograph main block (4E00-9FFF), the C, J, K, and V are show on the same row, while the U column is on the second row. If the first row is empty that row is collapsed.

TE.31. 33.5 Code Charts and lists of character names – Latin A7C0-A7C1

A7C0 LATIN CAPITAL LETTER THORN WITH DIAGONAL STROKE

A7C1 LATIN SMALL LETTER THORN WITH DIAGONAL STROKE

As documented in WG2 N4836R, this character has been in continuous use in typesetting Old English (OE) and Middle English (ME) texts since the time of Queen Elizabeth I in the mid-16th century. This tradition consistently uses a diagonal stroke, and is unrelated to the Nordic tradition of using a thorn with horizontal stroke for Old Norse (ON) texts. At present it is confusing for Anglicists how to represent the thorn with diagonal stroke in OE and ME texts as the standard only has thorn with horizontal stroke (A764/A765), and all fonts follow the glyph form shown in the code charts.

As the number of attestations of the printed letter thorn with diagonal stroke is an order of magnitude greater than the number of attestations of printed letter thorn with horizontal stroke, if the two letters were to be unified then the code charts should show the letter with the diagonal stroke. However, as thorn with horizontal stroke has already been encoded for more than ten years, making this change would cause unnecessary confusion for font designers, vendors and end users. Defining variation sequences for the two letters would not be a good solution either, as most scholars working on medieval languages will not understand how to apply a variation selector, or even appreciate that such a thing exists.

We maintain that the best solution is to encode a casing pair of letters for thorn with diagonal stroke. This way, users working on ON texts will be able to continue using A764/A765, whereas users working on OE/ME texts will be able to use A7C0/A7C1, and there will be no confusion as to what glyph form to use for what purpose, and the same font can be used for both OE/ME and ON texts.

Proposed change by U.K.

Keep A7C0 and A7C1 at the present positions.

If the two characters are removed from the draft standard then there will be a negative vote on the next ballot.

Noted

See also comment T1 from Ireland and TE.2 from US.

On principle the last part of the comment does not make much sense, a negative vote on a repertoire which is not there is at minimum an odd position.

TE.32. Code Charts and lists of character names – Latin A7D0-A7D9

A7D0 LATIN SMALL LETTER A WITH OVERCURL
A7D1 LATIN SMALL LETTER E WITH OVERCURL
A7D2 LATIN SMALL LETTER I WITH OVERCURL
A7D3 LATIN SMALL LETTER M WITH OVERCURL
A7D4 LATIN SMALL LETTER N WITH OVERCURL
A7D5 LATIN SMALL LETTER R WITH OVERCURL
A7D6 LATIN SMALL LETTER S WITH OVERCURL
A7D7 LATIN SMALL LETTER T WITH OVERCURL
A7D8 LATIN SMALL LETTER U WITH OVERCURL
A7D9 LATIN SMALL LETTER Y WITH OVERCURL

There is no need to encode atomic characters for each attested combination of letter with overcurl. As this mark is productive, other letters with overcurl may well be found at a later date, and would need separate encoding with this model.

We believe that the best solution is to encode a single combining diacritical mark for the overcurl.

In order to help font designers, we propose that Michael Everson should write a Unicode Technical Note (UTN) that illustrates how the overcurl should be drawn on all letters of the basic English alphabet (including those for which overcurl is not attested). With this UTN there is no longer any justification for encoding separate letters with overcurl.

Proposed change by U.K.

Remove A7D0..A7D9, and replace with 1ABF COMBINING OVERCURL in the Combining Diacritical Marks Extended block.

Accepted

See also comment T2 from Ireland and TE.3 from US.

Comment substantially identical to T2 from Ireland.

ED.33. 33.5 Code Charts and lists of character names – Newa 11460-11461

11460 NEWA SIGN JIHVAMULIYA
11461 NEWA SIGN UPADHMANIYA

The glyphs inside the dashed box for these two characters seem overlarge and out of proportion compared with the glyphs for other Newa characters.

Proposed change by U.K.

Reduce the size of the glyphs inside the dashed box for 11460 and 11461 in order to harmonize with the other characters in the Newa block.

Accepted in principle

Based on the editor receiving a new font for these 2 characters.

TE.34. 33.5 Code Charts and lists of character names – Zanabazar – 11A48-11A49

11A48 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER LA
11A49 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER SA

More evidence for the need of these two characters was provided in WG2 N4945R than was provided in the original Zanabazar Square proposals for 11A3A ZANABAZAR SQUARE CLUSTER-INITIAL LETTER RA. Although the evidence provided in WG2 N4945R was not perfect, due to the limited availability of Zanabazar Square texts, we believe that there is still a convincing need to encode 11A48 and 11A49.

Proposed change by U.K.

Keep 11A48 and 11A49 at their current positions.

Noted

See comment TE5 from US and its disposition.

ED.35. 33.5 Code Charts and lists of character names – Khitan – 16FE4

16FE4 KHITAN SMALL SCRIPT FILLER

The glyph shows the letters “KITS F” in a dashed box. “KSSF” for “Khitan Small Script Filler” would be better.

Proposed change by U.K.

Change glyph of 16FE4 to show the letters “KSSF” in a dashed box.

Accepted in principle

Based on the editor receiving a new font for these 2 characters.

TE.36. 33.5 Code Charts and lists of character names – Counting Rod – 1D379-1D37D

1D379 SOUTHERN SONG COUNTING ROD UNIT DIGIT FOUR

1D37A SOUTHERN SONG COUNTING ROD UNIT DIGIT FIVE

1D37B SOUTHERN SONG COUNTING ROD UNIT DIGIT NINE

1D37C SOUTHERN SONG COUNTING ROD TENS DIGIT FIVE

1D37D SOUTHERN SONG COUNTING ROD TENS DIGIT NINE

These characters were proposed by someone who is not an expert in traditional East Asian symbols, and who does not even read Chinese, so the proposal document (WG2 N4868) lacks sufficient detail and background discussion to properly evaluate the proposal. It is unclear whether the five proposed symbols are all that are needed for this system, and whether they could not be considered as unifiable variants of the corresponding Suzhou numbers.

We believe that these characters should be subject to further review by subject experts from China and elsewhere before they are accepted for encoding.

Proposed change by U.K.

Remove 1D379..1D37D pending further review and research.

Not accepted

Not reading Chinese should not be a criterion to disqualify such an encoding proposal. Many experts didn't 'read' the writing system or scripts they successfully proposed into the standard. Keeping in CD.2 will still provide additional time to review the encoding proposal and do research concerning these counting rod symbols.

Additional details about the proposal is available in

<http://www.unicode.org/L2/L2017/L2017085-counting-rod-std-var-seq.pdf>.

TE.37. 33.5 Code Charts and lists of character names – Geometric Shapes

1F7E0 LARGE ORANGE CIRCLE

1F7E1 LARGE YELLOW CIRCLE

1F7E2 LARGE GREEN CIRCLE

1F7E3 LARGE PURPLE CIRCLE

1F7E4 LARGE BROWN CIRCLE

1F7E5 LARGE RED SQUARE

1F7E6 LARGE BLUE SQUARE

1F7E7 LARGE ORANGE SQUARE

1F7E8 LARGE YELLOW SQUARE

1F7E9 LARGE GREEN SQUARE

1F7EA LARGE PURPLE SQUARE

1F7EB LARGE BROWN SQUARE

1F90D WHITE HEART

1F90E BROWN HEART

No evidence has been provided for existing use of these characters. Moreover, no evidence has been provided that users or vendors have requested that these characters be encoded. They have been proposed for encoding by the Unicode Emoji Subcommittee (ESC) because they think they would be a good idea for use with emoji to indicate a specific colour for a preceeding or following emoji.

This type of speculative encoding of characters on the basis that they would be a “good idea” is totally against UCS encoding principles, and should be firmly rejected. Before these characters are considered for encoding, at the very least the ESC should carry out in-depth market research to determine whether these characters would be accepted by emoji users to indicate the colour of a preceeding or following emoji (L2/18-208 page 1 suggests that they are not acceptable to emoji users, and would not be used for the purpose suggested by ESC).

Proposed change by U.K.

Remove 1F7E0..1F7EB and 1F90D..1F90E pending further evidence that they are required for encoding.

Not accepted

See comment T3 from Ireland and its disposition.

TE.38. 33.5 Code Charts and lists of character names – CJK Ext G – 30E21

UK submitted UK-01969 to IRG Working Set 2015, but as it was misprinted in the source (*Hànyǔ Dà Zìdiǎn*, 2nd ed., 2010) it was excluded from the CJK Ext. G submission. We have recently provided additional evidence for this character which verifies its glyph form (see http://appsrv.cse.cuhk.edu.hk/~irg/irg51/IRGN2308_UK-01969.pdf).

As this character completes the set of derived simplified characters in *Hànyǔ Dà Zìdiǎn* (2nd ed., 2010), it would be best to include it in Ext. G with the rest of the characters in this set

Proposed change by U.K.

Add UK-01969 (RS 140.9) between 30C5C (T13-3068) and 30C5D (KC-03602).

Not accepted

In <http://appsrv.cse.cuhk.edu.hk/~irg/irg51/IRGN2327WS2015EditorialReport.pdf> (IRG#51 editorial report, it was decided that:

A former WS2015 character (SN03617 UK-01969 𪛗) is accepted for inclusion in WS2017 as an exceptional case. (Also recorded in IRGN2328 Editorial Report on WS2017.)

Typically, IRG is not amenable at reinserting a character that was removed from an extension in an update of such extension. The UK experts are invited to submit this feedback to IRG#52.

TE.39. 33.5 Code Charts and lists of character names – CJK Ext G – 30E21

There is some confusion over the glyph form of 30E21 (GZ-1231301). The evidence shows {𪛗}, and this was the glyph form shown in the previous version of the Ext. G code chart. This was realised to be a mistake in the evidence, and so the character has been corrected to {𪛗} in the current code chart. However, some experts have questioned this correction.

Additional evidence from *Nán níng Píng huà Cí diǎn* 南寧平話詞典 [Dictionary of Nanning pinghua dialect] (Jiangsu Jiaoyu Chubanshe, 1997) page 175 confirms that {𪛗} is the correct glyph form:

【𪛗】timɿ 突然將腳或手抬起：你踩中我
腳喇，快啲～腳起來 || 壯語有 diem[tiː
m¹]這個詞，古壯字作“𪛗、𪛗、點”是抬
(腳)的意思

The RS for 30E21 is given as 157.9, but it should be 157.8 to match the correct glyph form.

Proposed change by U.K.

Do not change the glyph for 30E21.

Correct 30E21 (GZ-1231301) RS to 157.8.

Reorder 30E21 (GZ-1231301) between 30E12 (GHR-73955.06) and 30E13 (UK-02121).

Accepted in principle

However, to avoid further disruption in the Ext G code chart, this change will need to be validated by IRG#52 and if accepted by IRG experts it will be incorporated.

TE.40. 33.5 Code Charts and lists of character names – CJK Ext G – 300C6, 3029A, 30773, and 323C

The following characters have incorrect Radical/Stroke values (see IRGN2269_KR_Resp2R.pdf):

300C6 (KC-00229): RS 9.17 should be 9.16

3029A (KC-00729): RS 32.11 should be 32.12

30773 (KC-02200): RS 85.18 should be 85.19

323C (KC-04718): RS 196.12 should be 196.13.

Proposed change by U.K.

Correct 300C6 (KC-00229) RS to 9.16. No reordering required for 300C6 (KC-00229).

Correct 3029A (KC-00729) RS to 32.12. Reorder 3029A (KC-00729) between 302A6 (GZ-4921103) and 302A7 (UTC-01219).

Correct 30773 (KC-02200) RS to 85.19. Reorder 30773 (KC-02200) between 30774 (KC-05297) and 30775 (GHR-31928.04).

Correct 3123C (KC-04718) RS to 196.13. Reorder 3123C (KC-04718) between 31240 (GHR-84968.22) and 31241 (GHR-84976.08)..

Not accepted

IRG#51 editorial report at <http://appsrv.cse.cuhk.edu.hk/~irg/irg/irg51/IRGN2327WS2015EditorialReport.pdf> does not address these concerns. The editor has no generic issue with the stroke count and/or the re-ordering. However, all these CJK Ext G matters should be addressed as a whole by IRG.

TE.41. Page 2756, Annex A.1 Collections

Collections 2001 through 2007 are for CJK unified and compatibility ideographs in SIP and TIP (2007 = CJK Unified Ideographs Extension G).

Collection 2000 is for SIP, but there is no collection for TIP.

As CJK Unified Ideographs Extension G continues the 2000 series, we suggest redefining 2000 to cover SIP and TIP..

Proposed change by U.K.

Change Collection 2000 to “SIP and TIP” with code points 20000 through 3FFFD.

Accepted in principle

The exact code point ranges would be 20000-2FFFD and 30000-3FFFD.

ED.42. page 2785, Annex I.3, Table I.1

The IDS examples are not the same as given in the Unicode Standard, which may be confusing for users of the two standards.

Proposed change by U.K.

Change the IDS examples in Table I.1 to match the examples shown in the Unicode Standard 11.0 Figure 18-8.

Accepted

For reference the figure 18-8 is repeated below:

Figure 18-8. Examples of Ideographic Description Characters

U+2FF0	𠄐	U+4EC1	仁	→	𠄐イ二
U+2FF1	𠄑	U+5409	吉	→	𠄑士口
U+2FF2	𠄒	U+8857	街	→	𠄒彳圭子
U+2FF3	𠄓	U+58F9	壹	→	𠄓士一豆
U+2FF4	𠄔	U+56DE	回	→	𠄔口口
U+2FF5	𠄕	U+51F0	凰	→	𠄕几皇
U+2FF6	𠄖	U+51F6	凶	→	𠄖口乂
U+2FF7	𠄗	U+5321	匡	→	𠄗仁王
U+2FF8	𠄘	U+4EC4	仄	→	𠄘厂人
U+2FF9	𠄙	U+5F0F	式	→	𠄙弋工
U+2FFA	𠄚	U+8D85	超	→	𠄚走召
U+2FFB	𠄛	U+5DEB	巫	→	𠄛工从

ED.43. page 2788, Annex L, Guideline 1

“The name of an entity wherever possible denotes its customary meaning (for example, the character name: PLUS SIGN or the block name: BENGALI).”

BENGALI is probably not a good example to use, given the controversy over the naming of this block.

Proposed change by U.K.

Use a different example of a block name.

Accepted

ED.44. page 2828, Annex S.1.4.3

Very poor quality font images are used instead of coded characters.

Proposed change by U.K.

Use coded characters instead of images, with different fonts for the variant glyphs, or else use a single font and IVS sequences.

Accepted in principle

It is true that pictures are used for some part of these examples (6 pairs out of the 21 pairs) and could be improved by using fonts (although the current shapes are not ‘poor quality’), but the current status already reflects a significant change from earlier versions. At the same time, exact shapes are important to that annex and any changes need to be carefully analyzed by IRG experts. If a font was provided to the editor containing the required exact shapes, the corresponding font images could be incorporated.

USA: Negative

Technical comments:

TE.1. page 42, 26.2 Name formation

The USNB requests adding the following text to the paragraph in subclause 26.2 (which starts with the text “An entity name shall not contain”):

“An entity name shall not contain a HYPHEN-MINUS which is both preceded and followed by a SPACE character.”

Proposed change by US:

Add the new sentence as described.

Accepted

The rationale for that change is explained in <https://www.unicode.org/L2/L2018/18267r-char-name-restr.txt>.

TE.2. 33 Code charts and lists of character names – Latin Extended D – A7C0-A7C1

The USNB requests the removal of U+A7C0 LATIN CAPITAL LETTER THORN WITH DIAGONAL STROKE and U+A7C1 LATIN SMALL LETTER THORN WITH DIAGONAL STROKE. In the feedback document L2/18-286, the user community states that the angle of the stroke does not reflect a semantic distinction, and fonts are already being used with existing characters (U+A765 and U+A764) for both shapes. However, an annotation could be added, stating that the glyph variant with a diagonal stroke is widely used.

Proposed change by US:

Remove U+A7C0 LATIN CAPITAL LETTER THORN WITH DIAGONAL STROKE and U+A7C1 LATIN SMALL LETTER THORN WITH DIAGONAL STROKE.

If this comment, te.3, and te.4 are satisfied, the USNB vote will be changed to “Yes.”

Not accepted

See comment T1 from Ireland and TE.31 from UK.

Document L2/18-286 is also WG2 N5013.

The characters are kept on the general principles adopted for the transition between CD.1 and CD.2. This does not prevent US to make same comment on CD.2.

TE.3. 33 Code charts and lists of character names – Latin Extended D – Overcurl

The USNB requests the removal of ten Latin small letters with overcurl (U+A7D0..U+A7D9). Evidence showing meaningful orthographical distinction between letter and breve and letter with overcurl should be provided, in order to justify that these are not simply ligatures of letter and inverted breve.

Proposed change by US:

Remove the 10 letters.

If this comment, te.2, and te.4 are satisfied, the USNB vote will be changed to “Yes.”

Accepted

See comment T2 from Ireland and TE.32 from UK.

See disposition of comment for T2 from Ireland.

The 10 letters are removed. However, a combining character was proposed instead. The original proposal is N4907.

TE.4. 33 Code charts and lists of character names – Supplemental Symbols and Pictographs – 1F979 TROLL

The USNB requests the removal of U+1F979 TROLL. The UTC has never received a proposal for this character. The addition of such a pictographic character needs to be justified based upon appropriate criteria for encoding emoji.

Proposed change by US:

Remove U+1F979 TROLL.

If this comment, te.2, and te.3 are satisfied, the USNB vote will be changed to “Yes.”

Not accepted

See comment T5 from Ireland and its disposition.

The character is kept on the general principles adopted for the transition between CD.1 and CD.2. This does not prevent US to make same comment on CD.2.

TE.5. 33 Code charts and lists of character names – Zanabazar Square – 11A48-11A49

The USNB requests the removal of U+11A48 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER LA and U+11A49 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER SA. The characters are still lacking evidence of contrastive distinction in text.

Proposed change by US:

Remove U+11A48 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER LA and U+11A49 ZANABAZAR SQUARE CLUSTER-INITIAL LETTER SA.

Not accepted

See also comment TE.34 from UK.

The character is kept on the general principles adopted for the transition between CD.1 and CD.2. This does not prevent US to make same comment on CD.2.

TE.6. 33 Code charts and lists of character names – Khitan Small Script – 16FE4

The USNB requests the glyph for U+16FE4 KHITAN SMALL SCRIPT FILLER be changed so the abbreviation letters read “KSSF” instead of “KITSF”. “KSSF” is a more appropriate abbreviation for the name of the character.

Proposed change by US:

Change the glyph for U+16FE4 KHITAN SMALL SCRIPT FILLER as described.

Accepted in principle

Based on receiving an updated glyph.

TE.7. 33 Code charts and lists of character names – Adlam – 1E94B

Due to the urgent request from the Adlam user community and the planned publication of this character in Unicode 12.0, the USNB requests the addition of U+1E94B ADLAM NASALIZATION MARK in the CD. The full character proposal is located at: L2/18-282r.

Proposed change by US:

Add U+1E94B ADLAM NASALIZATION MARK.

Accepted

The editor has the glyph.

TE.8. 33 Code charts and lists of character names – Phags-Pa – A86D

Correct the glyph for U+A86D PHAGS-PA LETTER ALTERNATE YA as shown in L2/18-280 (WG2 N5012). There was an error in the code chart glyph; detailed background is provided in L2/18-280 (WG2 N5012).

Proposed change by US:

Correct the glyph for U+A86D PHAGS-PA LETTER ALTERNATE YA.

Accepted

The editor has the glyph.

Peter Constable comments (WG2 N5018)

The comments provided in WG2 N5018 are mostly editorial and concern useful clarification that can be conveyed in the text of the CD.

E.1 page vii, Foreword

The following text reflects the fifth edition, not the sixth, and needs to be updated.

This fifth edition of ISO/IEC 10646 cancels and replaces the fourth edition (ISO/IEC 10646:2014), which has been technically revised. It also incorporates ISO/IEC 10646:2014/Amd 1:2015 and ISO/IEC 10646:2014/Amd 2:2016.

This edition includes the following significant changes with respect to the previous edition:

- New scripts covered: Adlam, Bhaiksuki, , Marchen, Masaram Gondhi, Newa, Nushu, Osage, Soyombo, Tangut, and Zanabazar Square,
- Existing scripts significantly extended: Cherokee, CJK Unified Ideographs (Extension F),
- New Emoji symbols.

Change to:

This sixth edition of ISO/IEC 10646 cancels and replaces the fifth edition (ISO/IEC 10646:2017), which has been technically revised. It also incorporates ISO/IEC 10646:2017/Amd 1:2018 and ISO/IEC 10646:2017/Amd 2:2019.

other text as the editor deems appropriate.

Accepted

E.2 page 1, 1 Scope

The items in the bulleted list are delimited using comma “,”. However, several of the bullet items include expressions separated with commas. Normal editorial convention in such cases is for the higher-level boundaries to be separated using semi-colons. See clause 23 in ISO/IEC Directives Part 2, 8th edition, for an example that illustrates this.

Proposed change:

Replace commas at the end of bullet items with semi-colon.

Accepted

E.3 page 1, 1 Scope

In the bulleted list, the fourth bullet covers the BMP, and the fifth bullet covers the assigned supplementary planes. This fundamental distinction between the BMP and supplementary planes is anachronistic, a hold-over from when there were separate 10646-1 and 10646-2 standards.

Proposed change:

Merge the fourth and fifth bullets into one:

- specifies the assigned planes of the UCS: the Basic Multilingual Plane (BMP), the Supplementary Multilingual Plane (SMP), the Supplementary Ideographic Plane (SIP), the Tertiary Ideographic Plane (TIP), and the Supplementary Special-purpose Plane (SSP);

(Proposed text ends with semi-colon, as proposed in comment 2.).

Accepted

G.4 page 2, 3 Terms and definitions

Links are provided to terminology databases in IEC Electropedia and ISO OBP. Terms from 10646 do not appear to be included in either of these, however.

Proposed change:

None.

Noted

These links are required parts of the current ISO standard template.

E.5 page 3, 3 Terms and definitions – 3.10 code point

Add note:

Note 1 to entry — Code points in the UCS codespace are integer values. Throughout this document, UCS code points are cited in hexadecimal. UCS code points range from 0 to 10FFFF.

Accepted

E.6 page 3, 3 Terms and definitions – 3.12 code unit sequence

In note 1:

“... any type of code points.”

Change to:

“... any type of code point.”.

Accepted

E.7 page 3, 3 Terms and definitions – 3.12 code unit sequence

Note 1 refers to *types* of code points. However, this concept has not yet been introduced.

Proposed change:

Add at the end of note 1, “(See clause 6.3.)”

Accepted

T.8 page 4, Terms and definition – 3.13 Collection

The definition given is:

“numbered and named set of entities”

“Entities” is open-ended — it could include cities, unicorns, chemical formulas, etc. For the UCS context, something specific is intended, but there is no explanation of what that is. *Code points? Code point sequences?*

(Note: see related comment 15, below, for 3.25.)

Proposed change:

Add qualifiers in the definition or in a note (new note 1) clarifying what kinds of entities are included in UCS collections. Specific, proposed wording is not provided here since it’s not clear what is actually intended.

Accepted in principle

Entity was meant to mean either code point or sequences of code points. The new text is proposed:

collection

numbered and named set of entities made of code points or sequences of code points, the sequences conforming to Normalization Form C; code points lie within one or more identified ranges

Note 1 to entry – Non-extended collections do not contain sequences of code points (see also 3.25 for extended collection).

E.9 page 4, 3 Terms and definitions – 3.14 combining character

In note 1 (two occurrences):

“...non-combining graphic character...”

By 3.1, “non-combining graphic character” is the same as “base character”.

Proposed change:

replace “non-combining graphic character” with “base character”.

Accepted

E.10 page 4, 3 Terms and definitions – 3.15 combining class

Add note:

Note 1 to entry — See 20.2 for details on canonical ordering.

Accepted

E.11 page 4, 3 Terms and definitions – 3.17 composite sequence

In the Unicode Standard, the corresponding term is “combining character sequence”.

Proposed change:

Add “combining character sequence” as an alternate term for the same concept.

Accepted

E.12 page 4, 3 Terms and definitions – 3.17 composite sequence

The definition includes a cross-reference, “(see also 3.14)”. Cross references should be provided in notes, not in the definition. (Per 16.5.6 of ISO/IEC Directives Part 2, “The definition shall be written in such a form that it can replace the term in its context.”)

Proposed change:

Add a new note:

“Note 1 to entry — See also 3.14.”

Renumber subsequent notes.

Accepted

E.13 page 4, 3 Terms and definitions – 3.20 decomposition mapping

The terminology canonical equivalent and compatibility equivalent are used but are not defined in clause 3, or anywhere else in the document!

The definition of a term in clause 3 should not depend on terms that are not defined or explained within the doc. However, a thorough explanation would require details within chapter 3 of the Unicode Standard as well as UAX #15.

Proposed change:

Replace the definition with the following:

mapping from a character to a sequence of one or more characters

Note 1 to entry — Decomposition mappings are of two types: canonical decompositions, and compatibility decompositions. These are used in the derivation of various normalization forms (see 28). The code charts for various blocks include decomposition mappings and distinguish between the two types of mapping (see 33.3).

Accepted in principle

(see 28) should be (see 21).

E.14 page 5, 3 Terms and definitions – 3.24 encoding scheme

In note 1:

“... Some of the UCS encoding schemes have the same labels as the UCS encoding form. However, they are used in different contexts...”

Two issues:

- The definite article in “the UCS encoding form” assumes this is unique and implicitly-understood.;

- The antecedent of “they” is unclear: labels for encoding schemes and encoding forms are used in different contexts? Or encoding schemes and encoding forms themselves are used in different contexts?

Proposed change:

“... Some of the UCS encoding schemes have the same labels as UCS encoding forms. However, references to encoding schemes and encoding forms generally occur in different contexts...”

Accepted

E.15 page 5, Terms and definition – 3.25 Extended collection

The definition of extended collection in 3.25 and the description of non-extended collections in 3.13 are unclear. The definition in 3.25 states (emphasis added),

“collection for which the entities *can also* consist of sequences of code points that are in Normalization Form C”

This seems to imply that a non-extended collection must not include “sequences of code points that are in Normalization Form C”. That, in turn, seems to imply that a non-extended collection *can* include sequences of code points so long as they are not in Normalization Form C (including sequences that have proper sub-sequences that are in Normalization Form C).

It’s not clear if that is the actual intent, however. In 3.13, non-extended sequences are described as sets that

“... consist only of those coded characters whose code points lie within one or more identified ranges”

That description is, itself, vague, since *any* set of coded characters code be defined using code points that “lie within one or more identified ranges”, unless some constraint is imposed on “identified ranges”. But it does not seem to correspond in any clear way to the definition in 3.25.

Proposed change:

Give clearer definitions. Specific, proposed wording is not provided here since it is not clear what is actually intended.

Accepted in principle

With the proposed rewording of the term for Collection (see disposition of T8), it does not seem necessary to change the definition for Extended Collection:

extended collection

collection for which the entities can also consist of sequences of code points that are in Normalization Form C (NFC)

T.16 page 5, Terms and definition – 3.27 Format character

“character whose primary function is to affect the layout or processing of characters around it”

It is unclear whether the term defined here is intended to have a 1:1 correspondence with the Format basic type in the code point classification given in clause 6.3.1. As described in comment 40 below, the class of Format characters is somewhat complex: most are invisible controls affecting layout or processing of neighboring characters, but some produce a visible presentation, albeit one that involves a complex interaction with neighboring characters.

Proposed change:

“character whose primary function is to affect the layout or processing of characters around it, or that is presented in a complex, graphic interaction with neighboring characters”

Accepted

E.17 page 5, Terms and definition – 3.28 General Category

In note 1:

“Each value is defined as General Category property using a two-letter abbreviation in the Unicode Standard...”

Wording is unclear. Change to:

“Possible values are two-letter abbreviations defined for the General Category property in the Unicode Standard...”

Accepted

E.18 page 6, Terms and definition – 3.31 high-surrogate code point

“code point in the range D800 to DBFF reserved for the use of UTF-16”

Change to:

“code point in the range D800 to DBFF

“Note 1 to entry — Reserved for use in UTF-16 (see 9.3).”

Accepted

E.19 page 6, Terms and definition – 3.32 high-surrogate code unit

“16-bit code unit in the range D800 to DBFF used in UTF-16 as the leading code unit of a surrogate pair (see 9.3)”

Change to:

“16-bit code unit in the range D800 to DBFF and used in UTF-16

“Note 1 to entry — A high-surrogate code unit is used as the leading code unit of a surrogate pair. See also 3.40, 3.55 and 9.3.”

Accepted

E.20 page 6, Terms and definition – 3.34 ill-formed code unit sequence subset

Sets and subsets are not ordered, whereas a sequence is an ordered list of entities. The entities in a sequence can be described as coming from a set, but a sequence can include multiple instances of a given entity, whereas the set does not. Hence, the terminology “sequence subset” is odd and unclear.

This issue arises in the term being defined as well as in the definition.

Also, the definition has restrictive relative clauses introduced using “which”, not “that”.

Proposed changes:

- Change the term to “ill-formed code unit subsequence” (or “sub-sequence”).
- Change the definition to the following.

“non-empty subsequence of a code unit sequence X that does not contain any code unit that belongs to a minimal well-formed code unit subsequence of X

“Note 1 to entry — An ill-formed code unit subsequence cannot overlap with a minimal well-formed code unit sequence.”

Note: this term is only found in clause 3.61.

Accepted in principle

The following is better aligned with the latest Unicode core specification (V11, conformance clause D84a), changes shown in red:

“non-empty subsequence of a code unit sequence X that does not contain any code units that also belong to a minimal well-formed code unit subsequence of X

“Note 1 to entry — An ill-formed code unit subsequence cannot overlap with a minimal well-formed code unit subsequence.”

E.21 page 6, Terms and definition – 3.36 interworking

This term is not used anywhere else within the document, so it is not clear why the term is defined at all. Per ISO/IEC Directives Part 2, clause 16.5.4, only terms that are used within the document should be included in clause 3.

Proposed change: Delete this clause

Accepted

E.22 page 6, Terms and definition – 3.37 ISO/IEC 10646-1

“... the specification of the overall architecture and the Basic Multilingual Plane (BMP)”

Change to:

“... the specification of the overall UCS architecture and of the Basic Multilingual Plane (BMP)”

Accepted

E.23 page 6, Terms and definition – 3.39 low-surrogate code point

“code point in the range DC00 to DFFF reserved for the use of UTF-16”

Change to:

code point in the range DC00 to DFFF

Note 1 to entry — Reserved for use in UTF-16 (see 9.3).”

Accepted

E.24 page 7, Terms and definition – 3.40 low-surrogate code unit

“16-bit code unit in the range DC00 to DFFF used in UTF-16 as the trailing code unit of a surrogate pair (see 9.3)”

Change to:

16-bit code unit in the range DC00 to DFFF and used in UTF-16

Note 1 to entry — A low-surrogate code unit is used as the trailing code unit of a surrogate pair. See also 3.32, 3.55 and 9.3.

Accepted

E.25 page 7, Terms and definition – 3.44 plane

“subdivision of the UCS codespace consisting of contiguous 65 536 code points beginning at a multiple of 65 536 which can be identified by a number from 00 to 10”

Change to:

subdivision of the UCS codespace consisting of 65 536 contiguous code points beginning at a multiple of 65 536

Note to entry 1 — UCS planes can be identified by a hexadecimal number from 00 to 10.

Accepted

E.26 page 7, Terms and definition – 3.49 row

“subdivision of a plane consisting of contiguous 256 code points beginning at a multiple of 256 which can be identified by a number from 00 to FF”

Change to:

subdivision of a plane consisting of 256 contiguous code points beginning at a multiple of 256

Note to entry 1 — Within the context of a given plane, rows can be identified by a hexadecimal number from 00 to FF.

Accepted

E.27 page 8, Terms and definition – 3.52 Supplementary Multilingual Plane for scripts and symbols

The names of planes do not include a description of the characters or blocks within the plane.

Proposed change:

Change the term to “Supplementary Multilingual Plane” (i.e., remove “for scripts and symbols”).

Accepted

E.28 page 8, Terms and definition – 3.55 surrogate pair

“representation for a single character...”

Change to:

“UTF-16 encoded representation for a single supplementary-plane character...”

Accepted

E.29 page 8, Terms and definition – 3.59 unpaired surrogate code unit

“code unit in a code unit sequence...”

Change to:

“code unit in a UTF-16 code unit sequence...”

Also add a note:

Note 1 to entry — Any unpaired surrogate code unit constitutes an ill-formed code unit sequence.

Accepted

E.30 page 8, Terms and definition – 3.61 well-formed code unit sequence

“... and contains no ill-formed code unit sequence subset”

Change to:

“... and contains no ill-formed code unit subsequence”

See the related comment 20, above, on clause 3.34.

Accepted

T.31 page 9, 4.2 Conformance of information interchange

In list item a),

“... Clause 0...”

Proposed change:

Change to “Clause 6”.

Accepted

Typo, there is no clause 0, clause 6 was meant.

T.32 page 9, 5 General structure of the UCS

In paragraph 2,

“... from 0 to 10FFFF.”

Change to:

“... from 0 to 10FFFF (hexadecimal).”

Note: this change is not needed if the proposed change for clause 3.10 given in comment 5, above, is accepted.

Accepted in principle

This is an editorial comment, not technical. Comment for 3.10 was accepted, therefore no need to change this.

E.33 page 9, 5 General structure of the UCS

Second item of bulleted list after paragraph 2:

“The Supplementary Multilingual Plane for scripts and symbols...”

Change to:

“The Supplementary Multilingual Plane...”.

Accepted

T.34 page 10, 5 General structure of the UCS

In the paragraph after the bulleted list:

“The Tertiary Ideographic Plane (TIP, Plane 03) is reserved for ideographic characters and is currently empty.”

As of this CD, the TIP is no longer empty.

Proposed change:

- Delete this sentence from that paragraph.
- Insert a bullet item in the preceding bulleted list:
“• The Tertiary Ideographic Plane (TIP, Plane 03).”

Accepted

E.35 page 10, 5 General structure of the UCS

In the last paragraph:

“... coding space...”

Change to:

“... codespace...”

Accepted

E.36 page 11, 6.1 Structure

Figure 1 calls out Supplementary planes. This is anachronistic, a hold-over from when there were separate 10646-1 and 10646-2 standards. (See related comment 3, above.) While the term “supplementary plane” still is useful for 10646, its usefulness is specific to the UTF-16 encoding form and description of the surrogate code points used for UTF-16. In a general description of the structure of the code space, it is better to omit this.

Not accepted

The term supplementary is used so often in definition that showing it in a picture is still desirable.

E.37 page 12, 6.2 Coding of characters

“Each encoded character within the UCS codespace is represented by an integer between 0 and 10FFFF identified as a code point.”

Proposed change:

insert “(hexadecimal)” after “10FFFF”.

Note: this change is not needed if the proposed change for clause 3.10 given in comment 5, above, is accepted.

Accepted in principle

Comment for 3.10 was accepted, therefore no need to change this.

E.38 page 12, 6.2 Coding of characters

“When referring to characters within plane 00, the leading two digits may be omitted; for characters within planes 01 to 0F, the leading digit may be omitted, such as”

The structure of this sentence is such that the examples that follow seems to pertain only to planes 01 to 0F.

Change to:

“When referring to characters within plane 00, the leading two digits may be omitted; for characters within planes 01 to 0F, the leading digit may be omitted. For example:”

Accepted

E.39 page 12, 6.3.1 Classification

“UCS code points are categorized in basic types...”

Change “in” to “into”

Accepted

T.40a page 12, 6.3.1 Classification – Table 1

The stated General Category values for Format are Cf, Zl and Zp, and the brief description says,

“Invisible, but affects neighbouring characters”

The stated General Category values for Control are Cc, and the brief description says,

“Control functions consisting of a single code point”

There are some issues with this.

The first issue pertains to Zl and Zp: the code points assigned these values are single code points, and the characters are control functions: LINE SEPARATOR, PARAGRAPH SEPARATOR. Thus, the GC values seem to fit the description currently given for Control.

A comparison with corresponding descriptions in the Unicode Standard may be useful: it gives (Unicode 11.0, Chapter 2, Table 2-3) a different description for Control:

“Usage defined by protocols or standards outside the Unicode Standard”

A similar description could be equally applied to the description of Control in 10646 since, as noted in clause 11, the semantics for all C0 and C1 controls and for DELETE are specified in a different standard, ISO/IEC 6429. The LINE SEPARATOR and PARAGRAPH SEPARATOR control functions, however, are defined in 10646, not in ISO/IEC 6429. Thus, if the description for Control were similar to that used in Unicode, distinguishing functions defined outside rather than within the current document, then Zl and Zp would not fit that description for Control.

Proposed change:

Change the brief description for Control to:

“Control functions defined in ISO/IEC 6429”

or to:

“Control functions defined outside this document”

Accepted in principle

The first option is preferred: “Control functions defined in ISO/IEC 6429”.

T.40b page 12, 6.3.1 Classification – Table 1

The second issue pertains to the description for Format and characters that have General Category of Cf: Most Cf characters are invisible; for example:

- 061C ARABIC LETTER MARK
- 180E MONGOLIAN VOWEL SEPARATOR
- 200B ZERO WIDTH SPACE
- 200C ZERO WIDTH NON-JOINER
- 200D ZERO WIDTH JOINER
- 200E LEFT-TO-RIGHT MARK
- 200F RIGHT-TO-LEFT MARK
- etc.

However, at least one Cf character might be considered conditionally invisible — that is, invisible in some contexts, but given a visual presentation in other contexts:

- 00AD SOFT HYPHEN

Moreover, several other Cf characters always produce a visual presentation, albeit one that involves some complex interaction with neighboring characters; for example:

- 0600 ARABIC NUMBER SIGN
- 0601 ARABIC SIGN SANAH
- 0602 ARABIC FOOTNOTE MARKER
- 0603 ARABIC SIGN SAFHA
- 0604 ARABIC SIGN SAMVAT
- 0605 ARABIC NUMBER MARK ABOVE
- 06DD ARABIC END OF AYAH
- 070F SYRIAC ABBREVIATION MARK
- 08E2 ARABIC DISPUTED END OF AYAH
- 110BD KAITHI NUMBER SIGN
- 110CD KAITHI NUMBER SIGN ABOVE

At a minimum, then, “invisible” in the description for Format seems to be inappropriate.

It is also worth noting that the definition of format character in clause 3.27 mentioned “layout or processing”.

Proposed change:

Change the brief description for Format to:

“Affects layout or processing of neighboring characters, or has a complex graphic interaction with neighboring characters”

Accepted

T.41 page 12, 6.3.3 Format characters

As described in comment 40, the class of Format characters is complex in that some are invisible controls while others are not. The following is proposed as a replacement for the current text:

Format characters form a class of characters that affect or strongly interact with neighbouring characters.

Most format characters are invisible but affect the layout or processing of neighboring characters. For example:

061C	ARABIC LETTER MARK
200B	ZERO WIDTH SPACE
2062	INVISIBLE TIMES

Some format controls have a visible presentation, but one that involves a complex graphic interaction with neighboring characters. For example:

0600	ARABIC NUMBER SIGN
070F	SYRIAC ABBREVIATION MARK

Proposed change:

Format characters form a class of characters that affect or strongly interact with neighbouring characters.

Accepted

E.42 page 13, 6.3.5 Private use characters

“All code points of Plane 0F and Plane 10, except for FFFFE, FFFFF, 10FFFE, and 10FFFF are reserved for private use.”

Wording could be clearer. Change to:

“All code points of Plane 0F and Plane 10 — with the exception of noncharacter code points FFFFE, FFFFF, 10FFFE, and 10FFFF — are reserved for private use.”

Accepted in principle

Only the dash symbols are necessary for clarification:

“All code points of Plane 0F and Plane 10 — except for noncharacter code points FFFFE, FFFFF, 10FFFE, and 10FFFF — are reserved for private use.”

E.43 page 13, 6.3.7 Noncharacter code points

‘Code point FFFE is reserved for “signature”.’

The term “signature” is nowhere defined, and the meaning and purpose of this statement unclear. There should also be a cross-reference to clause 10, in which context signatures are relevant.

Proposed change:

Create a separate note for discussion of FFFE, with wording as follows:

NOTE – Code point FFFE is reserved for use as a “signature” for detecting correct byte order in UTF-16 or UTF-32 text data streams. Because the UTF-16 and UTF-32 encoding forms use 16-bit and 32-bit code units, but many processes handle data streams as byte sequences, the ordering of bytes within UTF-16 or UTF-32 code units strongly affects the interpretation of text data streams encoded in these encoding forms. The character FEFF ZERO WIDTH NO-BREAK SPACE, which has a minimal effect on the meaning or processing of text, can be included in a text data stream. If the bytes for this character were interpreted using the wrong byte order, then the bytes would be interpreted as the noncharacter code point FFFE. Since inclusion of this noncharacter code point is not expected in valid text content, the process would be able recognize the correct byte order. See also 10.

Accepted

E.44 page 13, 6.4 Naming of characters

“Some characters may have one or more alternate names called character name aliases which are correction of the original names. Additional rules to be used for constructing the names of characters are given in 26.”

Proposed change:

Some characters may have one or more alternate names, called character name aliases, which are corrections of the original names.

NOTE — Character name aliases, which are normative, should not be confused with informative aliases, which are other names for characters that may be used outside this document but that are not normative.

Additional rules to be used for constructing the names of characters are given in 26.”

Accepted

T.45 page 14, 6.5 Short identifiers for code points (UIDs)

The alternative form as stated in b) is:

“The four-to-five-digit form of short identifier shall consist of the last four to five digits of the six-digit form. Leading zeroes beyond four digits are suppressed.”

This is problematic in that, as stated, it allows for (e.g.) “2345” to be treated as a short form identifier for the code point 012345. While it states that leading zeroes are suppressed, it does not require that all non-zero digits must be maintained.

Proposed change:

“The four-to-five-digit form of short identifier shall consist of the last four to five digits of the six-digit form, with all non-zero digits kept but any leading zeroes beyond four digits suppressed.”

Accepted

E.46 page 14, 6.6 UCS Sequence Identifier

“The UCS Sequence Identifier includes at least two UIDs...”

Change to:

“A UCS Sequence Identifier must include at least two UIDs...”

Accepted in principle

‘must’ replaced by ‘shall’ as in: “A UCS Sequence Identifier shall include at least two UIDs...”

T.47 page 14, 6.6 UCS Sequence Identifiers

The first paragraph describes the elements within a USI as UIDs with “[t]he syntax for UID1, UID2, etc.... specified in 6.5.” The syntax in 6.5 allows for different forms — 017F, +017F, etc. However, the BNF given later in 6.6 only allows for one form for citation of UIDs: hex digits without a prefixed “U” or “+”.

If the intent is that a valid USI can use any of the valid forms for UIDs, then the BNF in clause given later in 6.6 is incorrect. But if the intent is that a valid USI can only use the forms for UIDs that exclude “U” and “+”, then this should be stated in the first paragraph.

If the different forms for UIDs are permitted in USIs, then (depending on the BNF) that could allow for USIs that mix different forms for UIDs; e.g., “<0041, U+0301>”. If use of different UID forms is valid, then a note can be added that a USI should consistently use the same UID form for all UIDs.

Proposed changes:

This depends upon what has been the intent.

- If alternate UID forms are to be permitted:
 - O Change the BNF in 6.5 to include “UID” as a label for the UID (“UID ::= ...”), then use “UID” in the BNF in 6.6 to replace both occurrences of “(xxxx|xxxxx|xxxxxx)”.
 - O Add a note recommending that citations for USIs consistently use the same citation form for UIDs.
- If alternate UID forms are not to be permitted: Append clarification on UID forms in the fifth sentence of the first paragraph, as follows:

“The syntax for UID1, UID2, etc. is specified in 6.5, with an added constraint that, within a USI, the UIDs must not include a “U”, “u” or “+” prefix.’

Accepted in principle

The second option is preferred (there is no use of alternate UID form in 10646), with ‘must’ replaced by ‘shall’.

T.48 page 15, 8.3 Selected subset

“A selected subset shall always automatically include the code points from 0020 to 007E.”

It’s unclear whether “shall always automatically include” is intended to say that 0020 – 007E are always implicitly included, or that this range must be explicitly included for a subset specification to be valid.

For example, suppose that a device is described as supporting a subset specified as “collection 2 (LATIN-1 SUPPLEMENT)”. Which of the following is the intent of 8.3?

- The subset specification is interpreted as comprising the code points {0020 – 007E, 00A0 – 00FF}.
- The subset specification is invalid (since collection 1 is not explicitly included in the specification).

The wording seems more likely to mean the former. If so, the following is proposed wording to replace the sentence quoted above:

Proposed change:

A selected subset shall always automatically and implicitly include the code points from 0020 to 007E, even if the corresponding collection, collection 1 BASIC LATIN, is not explicitly listed in the subset specification.

Accepted

E.49 page 15, 9.2 UTF-8

“It indicates the number of continuing octets...”

The antecedent of “it” is unclear: “first octet”? “code unit sequence”? “arbitrary location”?

Change to:

“The first code unit indicates the number of continuing octets...”

Accepted

Text was in 2nd sentence of the 4th bullet item

E.50 page 15, 9.2 UTF-8

“Table 2 specifies the bit distribution for the UTF-8 encoding form, showing the ranges of UCS scalar values...”

Table 2 doesn’t clearly show scalar-value ranges; it shows bit distributions, from which a reader could derive scalar-value ranges with some effort.

Proposed change:

Add a column on the left of Table 2 to show scalar-value ranges:

Scalar values	Scalar value bit distribution	1 st octet	2 nd octet	3 rd octet	4 th octet
0000 to 007F	00000000xxxxxx	0xxxxxx			
0080 to 07FF	00000yyyyyxxxxxx	110yyyyy	10xxxxxx		
0800 to D7FF E000 to FFFF	zzzzyyyyyyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
10000 to 10FFFF	000uuuuuzzzzyyyyyyxxxxxx	11110uuu	10uuzzzz	10yyyyyy	10xxxxxx

Accepted

T.51 page 16, 9.2 UTF-8

“Because of the well-formedness conditions specified in table 9.2, the following octet values are disallowed in UTF-8: C0-C1, F5-FE.”

The octet value FF should also be disallowed, but it is not mentioned.

Proposed change:

Change “F5-FE” to “F5-FF”.

Accepted

E.52 page 16, 9.3 UTF-16

“UTF-16 optimizes the representation of characters in the BMP which contains the vast majority of common use characters.”

The use of symbols in the SMP as emoji has to some degree shifted the balance of “common use” between BMP and SMP such that “vast majority” may be somewhat overstated.

Proposed change:

delete the word “vast”.

Accepted in principle

Replace ‘the vast majority of’ by ‘most’.

E.53 page 17, 9.3 UTF-16

Proposed change:

Add an extra column to Table 4 to show scalar-value ranges (as proposed for Table 2 in comment 50). Also, separate 1st and 2nd code units into separate columns (parallel to Table 2).

Scalar values	Scalar value bit distribution	1 st code unit	2 nd code unit
0000 to D7FF, E000 to FFFF	xxxxxxxxxxxxxxxx	xxxxxxxxxxxxxxxx	
10000 to 10FFFF	000uuuuuxxxxxxxxxxxxxxxx	110110wwwwwxxxxxx	110111xxxxxxxxxxx

Accepted

E.54 page 17, 9.4 UTF-32

Just as “UCS-2” is obsolete and anachronistic, the term “UCS-4” is now superfluous and only serves to create potential ambiguity.

Proposed changes:

Remove references to UCS-4 in the primary text but add a note for historical continuity. The combined changes would be to revise clause 9.5 as follows:

Clause 9.5 UTF-32

UTF-32 is the UCS encoding form that assigns each UCS scalar value to a single unsigned 32-bit code unit.

NOTE — Former editions of this document included “UCS-4” as an alternate term synonymous with “UTF-32”. Use of the term “UCS-4” to refer to this encoding form is deprecated.

Because surrogate code points are not UCS scalar values, UTF-32 code units in the range 0000 D800-0000 DFFF are ill-formed.

Accepted

E.55 page 17, 10 UCS Encoding schemes

The capitalization in the heading is inconsistent with conventions. (Compare with clause 9, “UCS encoding forms”.)

Change “Encoding” to “encoding”.

Accepted

E.56 page 17, 10.1 General (encoding schemes)

The discussion of signatures should cross-reference clause 6.3.7.

Proposed change:

Add at the end of paragraph 1, “(See also 6.3.7.)”.

Accepted

E.57 page 17, 10.3 UTF-16BE

“The UTF-16BE encoding scheme serializes a UTF-16 code unit sequence by ordering octets in a way that the more significant octet precedes the less significant octet (also known as big-endian ordering).”

A minor wording change is proposed:

The UTF-16BE encoding scheme serializes a UTF-16 code unit sequence by ordering the octets for each code unit such that the more significant octet precedes the less significant octet (also known as big-endian ordering).

Accepted

T.58 page 17, 10.3 UTF-16BE

Note: This comment and comments 60, 62 and 64 each pertain to discussion of signatures. Each proposes similar text added in clauses 10.3, 10.4, 10.6 and 10.7. An alternate approach would be to add an additional sub-clause 10.9 “Use of signatures” that contains information along the lines proposed by these comments.

This clause indicates how an initial octet sequence <FE FF> would be interpreted as ZERO WIDTH NO-BREAK SPACE and not as a signature, but there is no positive statement regarding octet sequences that are treated as a signature.

Note: see also comment 43 above.

Proposed change:

add a note after paragraph 2:

NOTE — Because the code point FFFE is a noncharacter code point (see 6.3.7), the octet sequence <FF FE> is not valid in the UTF-16BE encoding scheme. If a data stream is assumed to be using the UTF-16BE encoding scheme, an initial octet sequence <FF FE> would serve as a signature strongly suggesting that the assumption of UTF-16BE is invalid. If a data stream is assumed to be using the UTF-16 encoding form but the encoding scheme has not been specified, initial octet sequences <FE FF> or <FF FE> can be used as a heuristic: <FE FF> would strongly suggest that UTF-16BE can be assumed, while <FF FE> would strongly suggest that UTF-16BE should not be assumed. (See also 10.5.)

Accepted

E.59 page 17, 10.4 UTF-16LE

“The UTF-16LE encoding scheme serializes a UTF-16 code unit sequence by ordering octets in a way that the less significant octet precedes the more significant octet (also known as little-endian ordering).”

A minor wording change is proposed:

The UTF-16LE encoding scheme serializes a UTF-16 code unit sequence by ordering the octets for each code unit such that the less significant octet precedes the more significant octet (also known as little-endian ordering).

Accepted

T.60 page 17, 10.4 UTF-16LE

As for comment 58, there is no positive statement regarding octet sequences that are treated as a signature.

Proposed change:

Add a note after paragraph 2:

NOTE — Because the code point FFFE is a noncharacter code point (see 6.3.7), the octet sequence <FE FF> is not valid in the UTF-16LE encoding scheme. If a data stream is assumed to be using the UTF-16LE encoding scheme, an initial octet sequence <FE FF> would serve as a signature strongly suggesting that the assumption of UTF-16LE is invalid. If a data stream is assumed to be using the UTF-16 encoding form but the encoding scheme has not been specified, initial octet sequences <FF FE> or <FE FF> can be used as a heuristic: <FF FE> would strongly suggest that UTF-16LE can be assumed, while <FE FF> would strongly suggest that UTF-16LE should not be assumed. (See also 10.5.).

Accepted

E.61 page 18, 10.6 UTF-32BE

“The UTF-32BE encoding scheme serializes a UTF-32 code unit sequence by ordering octets in a way that the more significant octets precede the less significant octets (also known as big-endian ordering).”

A minor wording change is proposed:

The UTF-32BE encoding scheme serializes a UTF-32 code unit sequence by ordering the octets for each code unit such that the more significant octets precede the less significant octets (also known as big-endian ordering).

Accepted

T.62 page 18, 10.6 UTF-32BE

As for comment 58, there is no positive statement regarding octet sequences that are treated as signature.

Proposed change:

Add a note after paragraph 2:

NOTE — Because the code point FFFE is a noncharacter code point (see 6.3.7), the octet sequence <00 00 FF FE> is not valid in the UTF-32BE encoding scheme. If a data stream is assumed to be using the UTF-32BE encoding scheme, an initial octet sequence <00 00 FF FE> would serve as a signature strongly suggesting that the assumption of UTF-32BE is invalid. If a data stream is assumed to be using the UTF-32 encoding form but the encoding scheme has not been specified, initial octet sequences <00 00 FE FF> or <00 FF FE> can be used as a heuristic: <00 00 FE FF> would strongly suggest that UTF-32BE can be assumed, while <00 00 FF FE> would strongly suggest that UTF-32BE should not be assumed. (See also 10.8.)

Accepted

E.63 page 18, 10.7 UTF-32LE

“The UTF-32LE encoding scheme serializes a UTF-32 code unit sequence by ordering octets in a way that the less significant octets precede the more significant octets (also known as little-endian ordering).”

A minor wording change is proposed:

The UTF-32LE encoding scheme serializes a UTF-32 code unit sequence by ordering the octets for each code unit such that the less significant octets precede the more significant octets (also known as little-endian ordering).

Accepted

T.64 page 18, 10.7 UTF-32LE

As for comment 58, there is no positive statement regarding octet sequences that are treated as a signature.

Proposed change:

Add a note after paragraph 2:

NOTE — Because the code point FFFE is a noncharacter code point (see 6.3.7), the octet sequence <00 00 FE FF> is not valid in the UTF-32LE encoding scheme. If a data stream is assumed to be using the UTF-32LE encoding scheme, an initial octet sequence <00 00 FE FF> would serve as a signature strongly suggesting that the assumption of UTF-32LE is invalid. If a data stream is assumed to be using the UTF-32 encoding form but the encoding scheme has not been specified, initial octet sequences <00 00 FF FE> or <00 00 FE FF> can be used as a heuristic: <00 00 FF FE> would strongly suggest that UTF-32LE can be assumed, while <00 00 FE FF> would strongly suggest that UTF-32LE should not be assumed. (See also 10.8.)

Accepted

E.65 page 18, 11 Use of control function with the UCS

The notation used in escape sequences, such as “02/00” or “02/15” (as seen in clause 12.1), may be unfamiliar and confusing to readers. (Does this derive from ISO/IEC 2022?)

Proposed change:

Proposed change: Add a note after the fourth or fifth paragraph explaining the notation and its origin; re-number subsequent notes in this sub-clause.

Accepted

The proposed note is as follows:

NOTE 1 – The notation aa/bb (where ‘a’ and ‘b’ represent decimal digits 0 to 9) is used by ISO/IEC 2022 and ISO/IEC 6429 to indicate octet value where ‘aa’ can take values from 02 to 07 and ‘bb’ can take values from 00 to 15; for example, 02/00 represents the value 20 in hexadecimal notation and 07/15 represents the value 7E in hexadecimal notation.

E.66 page 19, 12 Declaration of identification of features

The wording “declaration of identification of features” is very unclear. Moreover, throughout this clause, “identification” is used inconsistently in ways that are sometimes very unclear and confusing. For example, “the identification of this document” (in 12.1) is not actually referring to an identifier for a document entity but rather is referring to encoding characteristics attributed to a document entity (viz., that the given document uses UCS for the encoded representation of text).

More appropriate uses of “identifier” might be “encoding form identifier” or “encoding scheme identifier”. But for some instances of “identifier” or “identification” in clause 12, “declaration” would be a more-appropriate term.

It is also noted that some parts of the current text use “designation”.

This comment proposes a change to the heading for clause 12. Other related comments pertaining to portions of sub-clauses will be provided below.

Proposed change:

replace the heading for clause 12 with the following:

“Specification of text-encoding attributes”

or:

“Specification of text-encoding characteristics”.

Not accepted

See disposition of comment E.68 below.

E.67 page 19, 12.1 Purpose and context of identification

In paragraph 1:

“Code unit sequences conforming to this document are intended to form all or part of a composite unit of coded information that is interchanged between an originator and a recipient. The identification of this document...”

The antecedent of “this document” in the second sentence is ambiguous. In general, this wording is used as a self-reference to this standard. That usage occurs in the first and last sentences of paragraph 1, and by comparing with the 4th edition, it appears that multiple instances of “this International Standard” have been replaced with “this document”, including in the second sentence. However, given the current wording of this paragraph, the occurrence in second sentence appears to be referring to the “composite unit of coded information”.

Proposed change:

This issue is intertwined with the issues raised in comment 66. A proposed change for this issue combined with the other issues is provided in the following comment.

Not accepted

See disposition of comment E.68 below.

E.68 page 19, 12.1 Purpose and context of identification

This comment pertains to and is a continuation of general issues raised in comment 66.

Proposed changes:

- Replace the heading for clause 12.1 with the following:
“Purpose and context for specification of text-encoding characteristics”

- Replace the content of clause 12.1 with the following — this also incorporates a change related to the issue raised in comment 67:

A declaration of the UCS as the encoded representation for text within coded information should also be available to the recipient, along with declarations of the encoding form and encoding scheme adopted by the originator, and possibly also declarations regarding any subsets of the UCS character repertoire that are used. The route by which such declarations are communicated to the recipient is outside the scope of this document.

However, some standards for interchange of coded information may permit, or require, that the coded representation of text be declared as part of the interchanged information, and moreover that specification of the text-encoding characteristics be declared using particular coded representations for those declared characteristics. Clause 12 specifies coded representations for declaration of various text-encoding characteristics applicable to the UCS. This includes declarations to specify encoding schemes, to specify graphic character subsets, or to specify control character subsets. Such coded representations provide all or part of a declarative data element, which may be included in information interchange in accordance with the relevant information-interchange standard.

For the contexts in which such declarations are used, it is assumed that more significant octets in a coded representation shall precede the less significant octets when serialized. For this reason, the only encoding schemes that can be specified using the coded representations provided here are UTF-8, UTF-16BE, and UTF-32BE according to the relevant encoding forms (UTF-8, UTF-16, and UTF-32 respectively).

If two or more of the text-encoding specification are present within a declarative data element, the order of those specifications shall follow the order as specified in Clause 12.

NOTE – An alternative method for specification of text-encoding characteristics is described in Annex N..

Not accepted

See also comment ED10 from Japan and its disposition.

The clause 12 and the substance of its sub-clauses have been mostly unchanged since the very beginning of ISO/IEC 10646 in 1993 with the following exceptions:

- 1) replacement of ISO/IEC 10646 to 'this International Standard', and then to 'this document',*
- 2) usage of new terminology better aligned with Unicode (such as replacing CC-data-element by code unit sequence),*
- 3) removal of deprecated implementation levels, reducing it to identification of encoding forms and schemes.*

The first replacement was derived from ISO publishing guidelines. This has resulting in confusion between reference to the standard and identification of use of this standard. This is impacting this clause as well as Annex N (for which a comment was made by Japan).

Fixing that issue is clearly worthwhile and can be done by reversing the last editing change (going back to 'ISO/IEC 10646' as appropriate. But changing the terminology used in this clause seems unwise as it is tightly connected with legacy standards such as ISO/IEC 2022 and ISO/IEC 6429. It may make the clause easier to read with users unfamiliar with these standards, but it will make the standard harder to use in their own context.

Following is a rewrite of sub-clause 12.1 (modified areas emphasized in bold):

12.1 Purpose and context of identification

Code unit sequences conforming to this document are intended to form all or part of a composite unit of coded information that is interchanged between an originator and a recipient. The identification of **ISO/IEC 10646** (including the encoding form and the encoding scheme) and any subset of the coding space that have been adopted by the originator should also be available to the recipient. The route by which such identification is communicated to the recipient is outside the scope of this document.

However, some standards for interchange of coded information may permit, or require, that the coded representation of the identification applicable to the code unit sequence forms a part of the interchanged information. Clause 12 specifies a coded representation for the identification of UCS and a subset of **ISO/IEC**

10646, and of a C0 and a C1 set of control functions from ISO/IEC 6429 for use in conjunction with **ISO/IEC 10646**. Such coded representations provide all or part of an identification data element, which may be included in information interchange in accordance with the relevant standard.

In the context of these identifications, because the more significant octets shall precede the less significant octets when serialized, the only encoding schemes that can be selected are UTF-8, UTF-16BE, and UTF-32BE according to the relevant encoding forms (UTF-8, UTF-16, and UTF-32 respectively).

If two or more of the identifications are present, the order of those identifications shall follow the order as specified in Clause 12.

NOTE – An alternative method of identification is described in Annex N.

E.69 page 20, 12.1 Purpose and context of identification

In paragraph 4:

“the order of those [specifications] shall follow the order as specified in Clause 12.”

It's not clear what is meant by “the order as specified in clause 12”. Does this mean that the required ordering for different kinds of declaration must corresponds to the order of sub-clauses in which the different kinds are described (e.g., that a specification for encoding scheme must precede a specification of a graphic character subset because clause 12.2 precedes clause 12.3)?

Proposed change:

Revise the wording to clarify the intended meaning. (Specific wording is not proposed since it's not clear what is intended.)

Accepted in principle

Re-reading all iterations of that clause since 1993, it seems clear that the order of the clauses was important. Therefore, the text can be rewritten as:

If two or more of the identifications are present, the order of those identifications shall follow the sub-clause order as specified in Clause 12.

NOTE – For example, the identification of the encoding scheme should precede the identification of the subset of graphic characters.

E.70 page 20, 12.2 Identification of a UCS encoding scheme

This comment pertains to and is a continuation of general issues raised in comment 66.

Proposed changes:

- Replace the heading for clause 12.2 with the following:
“Specification of a UCS encoding scheme”
- Revise paragraph 1 as follows:
“When the escape sequences from ISO/IEC 2022 are used, specification of a UCS encoding scheme (see Clause 10) defined by this document shall be by a specification sequence chosen from the following list:”
- In the note, replace “designation sequences” with “specification sequences”.

Alternately, “encoding-scheme identifier sequence” could be used instead of “specification sequence”

Not accepted

See disposition of comment E.68 above.

E.71 page 20, 12.3 Identification of subsets of graphic characters

This comment pertains to and is a continuation of general issues raised in comment 66.

Proposed changes:

- Replace the heading for clause 12.3 with the following:
“Specification of graphic character subsets”

- Revise paragraph 1 as follows (this does not reflect the issue raised in the following comment):
“When the control sequences of ISO/IEC 6429 are used, specification of graphic character subsets (see Clause 8) defined by this document shall be by a control sequence IDENTIFY UNIVERSAL CHARACTER SUBSET (IUCS) as shown below.”

Not accepted

See disposition of comment E.68 above.

T.72 page 20, 12.3 Identification of subsets of graphic characters

In paragraph 1:

“When the control sequences of ISO/IEC 6429 are used...”

Is “control sequences of ISO/IEC 6429” actually what is intended here? Or should this be “escape sequences of ISO/IEC 2022”?

Proposed change:

Reference the appropriate standard.

Not accepted

Again, re-reading all iterations of that clause since 1993, it seems clear that the control sequences of ISO/IEC 6428 (aka ECMA-48)) are used, and therefore the text is correct. For example, CSI mentioned in the sequence below is a CONTROL SEQUENCE INTRODUCER defined in clause 8.3.16 of that standard, and it is not specified in ISO/IEC 2022.

T.73 page 20, 12.3 Identification of subsets of graphic characters

In paragraph 1:

“... by a control sequence IDENTIFY UNIVERSAL CHARACTER SUBSET (IUCS) as shown below.”

This is followed by:

“CSI Ps... 02/00 06/13”

It’s not clear if “CSI” is referring to 6429 control function “009B CONTROL SEQUENCE INTRODUCER” (cited in clause 11). In any case, there is not clear continuity between the reference to IUCS in paragraph 1 and the elaboration that follows.

Proposed change:

Provide a clearer specification. (Specific changes are not proposed since it is unclear what precisely is intended.)

Accepted in principle

Again, re-reading all iterations of that clause since 1993, it seems clear that the control sequence is an IUCS defined as IDENTIFY UNIVERSAL CHARACTER SUBSET, and the first element of the sequence is the control function CSI. A possible improvement/clarification can be the addition of a note following the sequence:

When the control sequences of ISO/IEC 6429 are used, the identification of subsets (see Clause 8) specified by this document shall be by a control sequence IDENTIFY UNIVERSAL CHARACTER SUBSET (IUCS) as shown below.

CSI Ps... 02/00 06/13

NOTE – CSI is the short name for the control function CONTROL SEQUENCE INTRODUCER in ISO/IEC 6429.

E.74 page 20, 12.3 12.3 Identification of subsets of graphic characters

In paragraph 3:

“The parameters are to be taken from the subset collection numbers as shown in 9.”

It’s unclear what “9” refers to. Given the context involving subsets, it appears that clause 8 may perhaps be intended. It’s not clear that Clause 8 would be appropriate, however, since it allows for means of describing subsets that may not be amenable to representation in escape sequences. So, perhaps Annex A is what is intended.

(It appears that the reference to “9” was introduced in the CD for the 4th edition, and that prior to that the text referenced Annex A.)

Proposed change:

Provide the correct reference. If clause 8 is intended, use “Clause 8” (rather than simply “8”) for clarity.

Accepted in principle

It is not clear why the reference to Annex A was removed. As noted by the comment, clause 8 includes limited subsets that are not identifiable by escape sequences. Therefore ‘9’ will be replaced by ‘Annex A’.

E.75 page 20, 12.4 Identification of control function set

This comment pertains to and is a continuation of general issues raised in comment 66.

Proposed changes:

- Replace the heading for clause 12.4 with the following:
“Specification of control function set”
- Revise paragraph 1 as follows:
“When the escape sequences from ISO/IEC 2022 are used, the specification of each set of control functions (see Clause 11) of ISO/IEC 6429 to be used in conjunction with ISO/IEC 10646 shall be a specification sequence of the type shown below.”
- Revise the second sentence of paragraph 3 as follows:
“The specification sequences for these sets shall be”

Alternately, “control-function-set identifier” could be used instead of “specification sequence”.

Not accepted

See disposition of comment E.68 above.

E.76 page 21, 12.5 Identification of the coding system of ISO/IEC 2022

This comment pertains to and is a continuation of general issues raised in comment 66.

Proposed changes:

- Replace the heading for clause 12.4 with the following:
“Specification of the ISO/IEC 2022 coding system”
- Revise the first sentence of paragraph 1 as follows:
“When the escape sequences from ISO/IEC 2022 are used, specification of a return, or transfer, from UCS to the coding system of ISO/IEC 2022...”

Not accepted

See disposition of comment E.68 above.

G.77 page 19-21, 12 Declaration of identification of features

As suggested by comments 66 through 76, the current state of Clause 12 has several issues. It serves no purpose to provide a specification that is unclear and ambiguous. Many of the issues raised in these comments are very old and have remained through several editions without correction. If it has not been important through all this time to provide a clearer specification, then perhaps that is an indication that clause 12 is no longer important and can be removed

Accepted in principle

Removal of the clause is probably too extreme. Another solution could be to move it to Annex N (or a new Annex). If it was Annex N, that annex would contain first identification under ISO/IEC 2022 context, and second, identification using ASN. However, the current status is fine too.