telu **L2/20-105**

TO: **UTC**

FROM: **Deborah Anderson, Ken Whistler, Roozbeh Pournader, Lisa Moore, Peter Constable, and Liang Hai**[1]

SUBJECT: **Recommendations to UTC #163 April-May 2020 on Script Proposals**

DATE: **April 20, 2020**

The Script Ad Hoc group met on February 21, March 20, and April 17, 2020 in order to review proposals. The following represents feedback on proposals that were available when the group met. A table of contents is provided below.

I. EUROPE

1. Latin

1a. Extended IPA and VoQS

Document: L2/20-116 Expansion of the extIPA and VoQS – Miller and Ball

[1] Also participating were Ben Yang, Craig Cornelius, Michel Suignard, Manish Goregaokar, Ned Holbrook, Andrew Glass, Marek Jeziorek, Norbert Lindenberg, Steven Loomis, Patrick Chew, Sesh Sadisivam, Markus Scherer, and Lorna Evans.

Comments: We reviewed this proposal for 22 characters used to represent disordered speech. This proposal includes a set of 21 IPA extension characters and one Voice Quality Symbol (VoQS). The proposal combines characters from proposals seen at the January 2020 UTC (L2/20-038 and L2/20-039), with the addition of MODIFIER LETTER SMALL FENG DIGRAPH.

The following comments were noted:

- On page 2, replace "See Comment" with "See 'Comment on range of modifier letters' on page 4."
- In the next revision, provide a suggested names list annotation for U+00A1 INVERTED EXCLAMATION MARK, citing its use in IPA Extensions. (This is a carry-over request from earlier discussion at the Script Ad Hoc.)
- Work with Roozbeh Pournader to propose a names list annotation for U+A7F8 MODIFIER LETTER CAPITAL H WITH STROKE. (Background on the request is contained in L2/20-113.)
- For U+1D78 MODIFIER LETTER CYRILLIC EN, add a new "Script Property" section with the wording "Add Latin to the set of scripts in the ScriptExtensions property for U+1D78 MODIFIER LETTER CYRILLIC EN." The new section can come after the "Properties" section.
- Move COMBINING LEFT[/RIGHT] PARENTHESIS ABOVE LEFT[/RIGHT] to be before the COMBINING LEFT[/RIGHT] PARENTHESIS BELOW LEFT[/RIGHT] in the list of proposed characters on page 1 and the properties on page 3. This change will result in the same order of combining parentheses in the Combining Diacritical Marks Extended block (i.e., U+1ABB COMBINING PARENTHESES ABOVE and U+1ABC COMBINING DOUBLE PARENTHESES ABOVE come before U+1ABD COMBINING PARENTHESES BELOW).
- In answer to the question on page 2 about including a note on the order for interdental diacritics, we believe no note is necessary.
- Members of the Script Ad Hoc and others are encouraged to check the proposed code points.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 1a of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-116 .

1b. Modifier letters
Document: L2/20-117 Unicode request for modifier-letter support – Miller

Comments: We reviewed this document, which proposes modifier letters found in IPA, IPA-derived and Americanist notation, and in other uses.

The following comments were made during discussion:

- Discuss the issue surrounding the MODIFIER LETTER CYRILLIC EN, since the glyph resembles a U+1D45 MODIFIER LETTER CAPITAL H, which is an uppercase (not small caps) character in the Latin script. (See also "Script Property" comments above in section 1a)
- Use the code points in the new "Phonetic Extensions Supplement-A" block (U+10780…U+107BF) for characters outside the BMP. The new block has already been incorporated in the Roadmap.
- Provide any missing glyphs.

- Move the Properties and References (pp. 33-35) to before the figures (currently starting on page 6).
- Remove figures 72-73 and 85. The rationale for removing the figures is noted below:
    - The modifier *eszett* (p. 25, figures 72-73) appears to be a bad typographic substitution for a beta, because the author relied on limited font(s). More evidence is needed for this character.
    - The Greek conventions in figure 85 represent a different context from phonetic transcription: the Greek characters in the figure are closer to Latin scribal conventions and should be handled by rich text (i.e., styling of the text). A case would need to be made for them to be carried in plain text, if they were required (such as in a corpus database).
- Remove "ibid." on page 5 and repeat the reference.
- Modify "Comments on additional letters" to "Comments on additional letters – not proposed" or "Additional superscript letters for review – not proposed," so it is clear the characters are not being proposed.
- Consider including the proposed character name alongside the IPA name in the figures (i.e., on p. 15 "Modifier hook-top heng [MODIFIER LETTER SMALL HENG WITH HOOK]").
- Include a Proposal Summary form.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 1b of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-117.

1c. Click-letters
Document: L2/20-115 Unicode request for additional phonetic click letters – Miller and Sands

Comments: We reviewed this proposal to add seven phonetic symbols used for click consonants. The Script Ad Hoc reviewed an earlier version of this proposal.

The following comments were made during discussion of this revised proposal:

- For the triple pipe, we recommend employing U+2980 TRIPLE VERTICAL BAR DELIMITER, since it can be used in more general contexts (i.e., outside of math contexts), unlike U+2AF4 TRIPLE VERTICAL BAR BINARY RELATION and U+2AFC LARGE TRIPLE VERTICAL BAR OPERATOR.
- Provide a code point for LATIN LETTER SMALL CAPITAL TURNED K.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 1c of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-115.

1d. Ligature, IPA Retroflex, Hooks and Tails
Document: L2/20-125 Unicode request for a ligature, expected IPA retroflex letters and similar hooks and tails – Miller

Comments: We briefly reviewed this document, which proposed nineteen characters and provided evidence for them. We recommend generally keeping upper- and lowercase pairs of modern-use characters in the same plane. (At some point when the BMP becomes full, this practice may no longer be possible.)

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 1d of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-125.

1e. Mapping of extIPA and modifier phonetic characters
Document: L2/20-118 Proposed mapping of extIPA and modifier phonetic characters – Miller

Comments: We briefly reviewed this document that contains a chart with the tentative locations of proposed phonetic characters. The author explained to the Script Ad Hoc that the characters below the chart are not yet assigned code points and the greyed boxes are for characters that have been published in Unicode 13.0. The author is planning to put the more important and common characters in the BMP and placing the rarer characters in the SMP.

The following comments were noted:

- We recommend assigning characters for U+A7C0 and U+A7C1.
- If possible, avoid splitting case pairs across different planes.
- The "Supplementary plain" [sic] should be renamed "Phonetic Extensions Supplement A"

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 1e of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-118.

II. AFRICA

2. Egyptian Hieroglyphs
Documents:
L2/20-068R Revised draft for the encoding of an extended Egyptian Hieroglyphs repertoire, all Groups (A to Z and AA) – Suignard (Appendix database as PDF)
L2/20-123 Zoom Meeting on Egyptian Hieroglyphs (April 8, 2020) – Anderson

Comments: We reviewed this revised draft of the extended Egyptian Hieroglyph repertoire, now covering all Gardiner groups (A to Z and AA). A PDF database with phonetic values is also now available. The total number of additions in the proposal is 6,826; including the currently encoded Egyptian Hieroglyphs (1,071 characters) the total comes to 7,897 characters.

The following points were raised during discussion:

- The new additions will require adjustment in the Roadmap for Egyptian Hieroglyphs Extended-B so the block extends from U+14680..U+151FF.

- A number of issues are still outstanding for the proposal author:
    - How should characters appearing within a container be handled? Michel Suignard can create a list of those elements that appear inside a container for font designers. Andrew Glass agreed that handling the enclosing signs dynamically within the font was a good approach.
    - In some cases, it can be difficult to identify the exact elements of a hieroglyph, which in turn affects the taxonomy.
- We recommend the author provide a guide identifying the key issues for UTC members. The guide should also discuss the principles used in deciding which characters to include (or not) and why. This latter goal will help UTC members understand the goals behind the project and make it clear that the encoding decisions are not arbitrary.
- The author would like to proceed on the repertoire soon. Having a stable, citable repertoire of characters would be useful for Egyptologists.
- The document by Anderson (L2/20-123) summarizes the discussion with experts in Egyptian hieroglyphs. It reflects concerns voiced at earlier meetings, such as in L2/19-314 (i.e., how will students pick the correct character, desire for decomposed encoding of characters that will rely on fonts to create the glyphs, concern over the number of large number of variants, etc.). The comments in L2/20-123 don't always square with the results of the London 2018 meeting of experts (L2/18-237), which had agreed:

    > It is advantageous to the field of Egyptology to include a large set of additional characters, including those in Hieroglyphica and other sources, with the responsibility on the user to select the correct character.

    It might be useful to have Andrew Glass to be on a future call with Egyptologists to explain the limits of the current font technology. Also, having the proposal author provide examples of how subsetting can be done and a demo of UniHan-like database for Egyptian Hieroglyphs would likely be helpful for experts.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the Roadmap Committee to request the block range for Egyptian Hieroglyphs Extended-B be modified, so the block extends from U+14680..U+151FF. Reference: Section 2 of L2/20-105 (Script Ad Hoc Recommendations).

III. MIDDLE EAST

3. Arabic
3a. Arabic Honorific
Document: L2/20-081 Proposal to encode an Arabic honorific used in Christian texts – Pournader and Evans

Comments: We reviewed this request for one Arabic honorific character, which was found in Christian religious texts. The evidence appears solid. The proposed code point, U+FDCF, is in the Arabic Presentation Forms-A block, which already includes recently approved Arabic honorifics. The location seems reasonable. Lorna Evans has agreed to provide a font.

Recommendation: We recommend that the UTC approve the following:
SAH-UTC163-R1: The UTC accepts U+FDCF ARABIC LIGATURE SALAAMUHU ALAYNAA for encoding in a future version of the standard. Reference: L2/20-081.

3b. Arabic Tail Character
Document: L2/20-071R Proposal to encode an Arabic tail character used for abbreviation – Pournader and Izadpanah

Comments: We reviewed this proposal for one Arabic character, ARABIC VERTICAL TAIL, which was used in early Persian typography as an abbreviation mark.

The character only appears in books printed by a specific press in Tehran in the 19th century. The character probably survived for only a short while, then disappeared. It is similar to a character proposed in 2001 (L2/01-095), U+FE73 ARABIC TAIL FRAGMENT, which was encoded for compatibility with legacy character sets and is non-joining. The proposed character, however, should be encoded as a normal shaping character. The examples show it as a right-joining character that appears after *sad* or *ain*. Future research could reveal its appearance after other letters.

The proposal appears to be well-formed and the code point, U+088E, acceptable. This character is proposed for inclusion in the new Arabic Extended-B block (U+0870..U+089F). We do not recommend that it be encoded in one of the Arabic Presentation Forms blocks, because this is a shaping character which likely was used productively in combinations with other letters. Lorna Evans has agreed to assist on a font.

Recommendation: We recommend that the UTC approve the following:
SAH-UTC163-R2: The UTC accepts U+088E ARABIC VERTICAL TAIL for encoding in a future version of the standard. Reference: L2/20-071R.

3c. Indonesian Orthography of Quran
Document: L2/20-089 Proposal to Encode Characters from Indonesian Orthography of Quran – Syarifuddin

Comments: We reviewed this proposal for five Quranic characters used in Indonesian orthography. The characters are comparable to those Quranic annotation characters from U+08DA..U+08DF.

The evidence is solid and the names and code points acceptable. The proposed location of the characters is in the Arabic Extended-B block. Note that this block is filling up and the next Arabic block will be on the SMP.

The recommended annotation for U+0653 ARABIC MADDAH ABOVE should be reviewed by the names list editor; the character is common so an annotation may not be necessary. Deborah Anderson is requesting a font from the proposal author, but Lorna Evans will act as a back-up.

Recommendations: We recommend that the UTC approve the following:
SAH-UTC163-R3: The UTC accepts the following five characters for encoding in a future version of the standard:
U+0898 ARABIC SMALL HIGH WORD AL-JUZ
U+0899 ARABIC SMALL LOW WORD ISHMAAM
U+089A ARABIC SMALL LOW WORD IMAALA
U+089B ARABIC SMALL LOW WORD TASHEEL
U+089C ARABIC MADDA WAAJIB.
Reference: L2/20-089.

We also recommend the UTC make the following disposition:
Assign an AI to the names list editor to review the proposed annotation for U+0653 ARABIC MADDAH ABOVE on page 3 of L2/20-089 and discussed in section 3c of L2/20-105 (Script Ad Hoc Recommendations).

3d. Kalasha Language
Document: L2/20-091 Proposal to include Kalasha Language alphabets – Rehmat Aziz Khan Chitrali

Comments: We reviewed this proposal for nine Arabic letters used to write the Kalasha language of Pakistan. Kalasha is presently written in the Arabic and Latin scripts. The author, a linguist and researcher, has created a keyboard and a Kalasha primer that uses these characters.

The following comments were raised during discussion:

- Remove the following two characters:
  - ARABIC LETTER NOON WITH ARABIC MARK NOON GHUNNAH ABOVE. This character can already be represented by the sequence <U+0646, U+0658>.
  - ARABIC LETTER LAAM WITH SMALL TUEY ABOVE. This character has already been approved and appears in Unicode 13.0 as U+08C7 ARABIC LETTER LAM WITH SMALL ARABIC LETTER TAH ABOVE.
- Those characters with sequences that include “U+0615” should be separately encoded as atomic units (except the seventh character, which is already encoded). U+0615 is a Quranic annotation sign and is not generative as a diacritic.
- Remove the “code point” column from the chart on page 1.
- Rename the characters as follows (or, alternatively, use names “…WITH SMALL **ARABIC LETTER** TAH ABOVE”):
  ARABIC LETTER ALEF WITH SMALL TAH ABOVE
  ARABIC LETTER TCHEH WITH SMALL TAH ABOVE
  ARABIC LETTER JEEM WITH SMALL TAH ABOVE
  ARABIC LETTER SHEEN WITH SMALL TAH ABOVE
  ARABIC LETTER JEH WITH SMALL TAH ABOVE
  ARABIC LETTER WAW WITH SMALL TAH ABOVE
  ARABIC LETTER YEH WITH SMALL TAH ABOVE
- Provide examples of the seven proposed characters above in print, as found in publications by the proposal author as well as by others. The characters should be shown to be in use by a group of people, not one person or a very small number of people.

- Is there any Kalasha language authority who supports these requests? If so, mention this in the proposal.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 3d of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-091.

3e. Luri
Document: L2/20-092 Proposal to include four Luri alphabets – Mogoei and Shaikh

Comments: We reviewed this proposal for four characters. It is based on the revised proposal L2/18-061, which had requested the following three characters:

Shape	Code Point	Name
	U+08C5	ARABIC LETTER ALEF WITH INVERTED V ABOVE
	U+08C6	ARABIC LETTER KEHEH WITH INVERTED V ABOVE
	U+08C7	ARABIC LETTER LAM WITH INVERTED V ABOVE

This proposal requests the following four characters:

Shape	Code Point	Name
	U+0894	ARABIC LETTER ALEF WITH SUKUN ABOVE
	U+0895	ARABIC LETTER ALEF WITH SMALL INVERTED V ABOVE
	U+0896	ARABIC LETTER KEHEH WITH SMALL INVERTED V ABOVE
	U+0897	ARABIC LETTER LAAM WITH SMALL INVERTED V ABOVE

The Script Ad Hoc had commented (L2/18-168) on the 2018 proposal (L2/18-061) and recommended that the first character, ARABIC LETTER ALEF WITH SUKUN ABOVE, be represented by the sequence <U+0627 ARABIC LETTER ALEF, U+0652 ARABIC SUKUN>. (This character was removed in the revised proposal, L2/18-061R). (The Script Ad Hoc had also suggested in L2/18-168 that ARABIC LETTER KEHEH WITH SMALL INVERTED V ABOVE be represented with U+063D ARABIC LETTER FARSI YEH WITH INVERTED V, but this recommendation was incorrect.)

In order to have enough information to adequately evaluate these characters, the Script Ad Hoc recommends the following:

- Describe the classification of the different dialects of the Luri language as understood by M. Mogoei.
- In which regions of Iran are the above dialects spoken? Is there a map of these dialects that M. Mogoei agrees with?
- Are all of the suggested alphabet letters used in each of these dialects? If not, which letters are used in which dialects?
- Clarify the differences in IPA and phonetic values below:

- For ٱ, the proposal has [æ], Omniglot has for Northern Luri [ʔa/ æ], and Figure 6 has [e] and [æ]

- For ڪ, the proposal has [kʔ], but is not listed in Omniglot, and figure 6 has:

ڪ ک	kʔ	(ک لری جنوبی) در فارسی وجود ندارد	ķ	ڪوچو – ڪینی

What does the k with diacritic indicate? How does it relate to [kʔ]?

- For ڵ: the proposal has [lʔ]. Omniglot gives ɮ (voiced alveolar lateral fricative), and figure 6 has:

30	ڵ ل	lʔ	(لام فیلی) در فارسی وجود ندارد	ɬ	توڵ - کیسڵ

Cf. the following from http://www.iranicaonline.org/articles/lori-dialects

> The NLori dialects have developed a strident lingual, /ɬ/, produced by bringing both the tip and sides of the tongue into contact with the alveolar ridge and allowing breath to escape around the tip of the tongue. This results in a hissing sound with a palatalized quality: *pił* "money," *łöj* "lip." Bak̲, Mam have [w] for Standard Pers. [v]

- Is there a Luri language authority in support of these requests? If yes, who are the experts in charge of it?
- Are there any other Luri orthographic traditions? If yes, how popular are they? Which of the orthographies have dictionaries published in them?
- Provide examples from printed materials and signage, including any texts authored by individuals, other than the co-author of the proposal. This will demonstrate to the UTC that the letters are not being used by a very small number of people.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 3e of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-092.

3f. Comments** on L2/19-306 **Arabic additions for Quranic orthographies
Reference: Error report

Comments: The following is an Error report that was submitted to Unicode in December 2019, with responses from the Script Ad Hoc.

> Date/Time: Mon Dec 9 18:19:01 CST 2019; > Name: Eduardo Marín Silva; Opt Subject: Three observation on the proposal of Quranic marks (L2/19-306)

> These observations are not contained in the recent feedback document by Azzeddine (L2/19-393). These are independent of that document.

> 1. Confusability. The document does not discuss the possibility of confusability that poses the character ARABIC ROUNDED HIGH STOP WITH FILLED CENTRE (06EC). This applies to the new forms of Alef with a large round dot at the top, as well as the new proposed combining character ARABIC LARGE ROUND DOT ABOVE.

> **Comments from Script Ad Hoc:**
> Noted. In our view, the spoofing risk is low for Quranic texts.

> 2. Name modification. It is clear from looking at the attestations of the proposed character: ARABIC ROUND DOT INSIDE LARGE CIRCLE BELOW, that the glyph is a normal sized dot enclosed in a large circle, but a large dot enclosed in an even larger circle. A name that better reflects this fact would be ARABIC "LARGE" ROUND DOT INSIDE LARGE CIRCLE BELOW, in my opinion.

> **Comments from Script Ad Hoc:**
> Based on the example for character #29 on page 16 of L2/19-306, (= U+08D2 on page 27), we agree that a name change for U+08D2 ARABIC ROUND DOT INSIDE LARGE CIRCLE BELOW to ARABIC **LARGE** ROUND DOT INSIDE CIRCLE BELOW makes sense. The proposed name change follows the pattern found in the two characters U+08CE ARABIC LARGE ROUND DOT ABOVE and U+08CF ARABIC LARGE ROUND DOT BELOW. Note that U+08D2 was approved by the UTC at the October 2019 meeting.

> 3. Missing pedagogical symbols. On page 17 of L2/19-306 (it appears once again on page 25), on the top figure; there is evidence of two new proposed marks: ARABIC LARGE ROUND DOT BELOW and ARABIC (LARGE) ROUND DOT INSIDE LARGE CIRCLE BELOW. However, on the same figure, they also appear on the baseline, as spacing characters like pedagogical symbols. These were not proposed as separate characters, which is inconsistent with the approved character ARABIC SYMBOL WASLA ABOVE, that follows a very similar rationale. Perhaps the spacing version of the first one can be unified with ARABIC BASELINE ROUND DOT, but there would still be a missing spacing version of the latter.

> **Comments from Script Ad Hoc:**
> To create a spacing version of the dot characters, users can use a non-breaking space before the character(s). If the author wants to write a proposal and justify separately encoding them, he is welcome to. In our view, they are not necessary.

Recommendations: We recommend that the UTC approve the following:
SAH-UTC163-R4: The UTC changes the name for U+08D2 from ARABIC ROUND DOT INSIDE LARGE CIRCLE BELOW to ARABIC **LARGE** ROUND DOT INSIDE CIRCLE BELOW. Reference: Section 3f of L2/20-105 (Script Ad Hoc Recommendations).

We recommend the UTC make the following disposition:
The UTC forwards the comments on items #1 and #3 in section 3f of L2/20-105 (Script Ad Hoc Recommendations) to the author of Error Report on L2/19-306 Arabic additions for Quranic orthographies.

IV. SOUTH AND CENTRAL ASIA

4. Ahom

Action Item 153-A126: Specify the order of Ahom Vowel Signs. See L2/17-364 PRI Feedback from David Corbett, dated September 27, 2017 – Anderson, Pournader, Glass, Constable, and Hosken

Comment from Corbett:

> An Ahom consonant may take multiple vowel signs, all of which have ccc=0. The Unicode Standard does not say what order they should be encoded in. The proposal (L2/12-309R) recommends an order, but contradicts itself: on page 2, it says U+1172A AHOM VOWEL SIGN AM should precede U+11724 AHOM VOWEL SIGN U, but on page 3, it gives the opposite order. It is therefore unclear what the intended order is.

Comments:
Norbert Lindenberg is in contact with Stephen Morey (who is in touch with the user community) and Andrew Glass. Norbert is investigating character sequences that were missed in the original proposal. The outcome will be reported at a future Script Ad Hoc and may include proposing new wording to the *TUS* block intro for Ahom, property changes, updates to Universal Shaping Engine, and/or keyboard layout specs.

Recommendation: We recommend the UTC make the following disposition:
The UTC forwards the comments in section 4 of L2/20-105 (Script Ad Hoc Recommendations) to David Corbett.

---

5. Brahmi

Documents:
L2/20-069 Proposal to Annotate Brahmi Sign Anusvara - Rajan and Sharma
Related documents:
L2/19-402 Proposal to Encode 6 Characters in the Brahmi Block – Vinodh Rajan and Shriramana Sharma
L2/20-037 Comments on L2/19-402 – Glass

Comments: We reviewed this proposal to unify OLD TAMIL VIRAMA with BRAHMI SIGN ANUSVARA, adding an annotation to the BRAHMI SIGN ANUSVARA to denote the Tamil-Brahmi usage.

The January 2020 UTC had approved five of the six Tamil Brahmi characters proposed in L2/19-402, but did not accept the OLD TAMIL VIRAMA. The discussion at the January UTC focused on whether the BRAHMI SIGN ANUSVARA could be used as the virama, since the *anusvara* is not used in Old Tamil. Another option might be using the currently encoded BRAHMI VIRAMA, but with an alternate shape. However, the earlier Script Ad Hoc recommendations in L2/20-046 noted the difference in behavior between Brahmi and Old Tamil viramas – one is conjunct-forming, and the other a pure killer.

Comments that arose during discussion include:

- The position of both the Old Tamil virama and Brahmi *anusvara* varies. The default position for Brahmi *anusvara* is on the top, and the Old Tamil virama is on the right.
- The signs are functionally different, but to some members of the Ad Hoc, the difference in properties was not deemed a strong enough argument to prevent unifying the Old Tamil virama with the *anusvara*.

Recommendation: We recommend the UTC make the following disposition:
Remand L2/20-069 Proposal to Annotate Brahmi Sign Anusvara back to the Script Ad Hoc for further discussion, inviting Andrew Glass to present his viewpoint. Reference: Section 5 of L2/20-105 (Script Ad Hoc Recommendations).

---

6. Gurmukhi
Documents:
L2/20-060 Feedback on Gurmukhi Bindi Before Bihari (L2/18-319, L2/19-167, L2/19-283) – Sarabveer Singh
L2/20-076 Summary of email discussion on Gurmukhi BINDI – Anderson

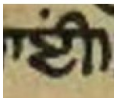
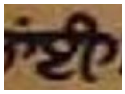
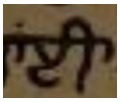
Comments: We reviewed the two documents.

The document by Sarabveer Singh (L2/20-060) states his view that *bindi* occurring before *bihari* is stylistic, and hence should be supported in fonts as a font feature.

The second document (L2/20-076) summarized an email thread between Kulpreet Chilana, Irvanjit Singh, and Manvir Singh about Gurmukhi *bindi* before *bihari*. The document includes 22 examples (pp. 6-9) of the same Gurmukhi text written by different scribes. On page 1, Kulpreet Chilana offers different options on how to handle *bindi* before *bihari* if it can be shown that *bindi* (and *tippi*) are used consistently in the same position. Irvanjit Singh and Manvir Singh considered the following options to be acceptable: (1) Encode LEFT SIDE BINDI and LEFT SIDE TIPPI as separate code points ("potentially with varying character precedence and rendering exceptions?") or (2) use variation selectors with *bindi* and *tippi* to render left-side variants of these characters.

The following comments were noted:
- §23.4 of *TUS* states "A variation sequence always consists of a base character or a spacing mark (gc = Mc) followed by a single variation selector [VS] character." Hence, a VS can't apply to a *bindi* (gc=Mn) or a *tippi* (gc=Mn), if one is adhering to the Unicode Standard.
- The exact grammatical purpose of the *bindi* dot and its meaning are still unclear. As a result, the requirement of representing *bindi* before *bihari* in plain text has not yet been clearly shown.
- Could the *bindi* have migrated from the previous syllable? If so, this could explain a different pronunciation, but a different analysis is needed. Note the left-ward placement of the *bindi* in #8 and #17:

Expected placement (#5) Leftward placement (#8) (#17)

- The authors in L2/20-076 mention the importance of being able to cite the religious text in social media, but representing *bindi* before *bihari* remains an obstacle.
- Can Unicode provide guidance on what is the best way to achieve the *bindi* before *bihari* in fonts and non-specialized software? Or is a solution beyond the scope of Unicode and better suited to a solution with OpenType?

Recommendation: We recommend the UTC make the following disposition:
Remand the Gurmukhi *bindi* discussion back to the Script Ad Hoc for further discussion, inviting interested parties to participate. Reference: L2/20-060 and L2/20-076 and Section 6 of L2/20-105 (Script Ad Hoc Recommendations).

Note: See also section ***17.*** **i**n the Public Review Feedback at the end of this document for feedback on Gurmukhi.

---

7. Mongolian
7a. Mongolian proposal by Z Team/ Chuck Namujila
Documents:
MWG/4-N5 The Simplified Mongolian Solution - Unicode – Z Team / Chuck Namujila
MWG/4-N4 Preliminary proposal for encoding the Mongolian Usgiin Solution in the SMP of the UCS File No: File-2 – Chuck Namujila

Comments: We discussed Chuck Namujila's presentation "Simplified Mongolian Solution – Unicode", which he gave to the Script Ad Hoc. The presentation was a summary of the problems with the current Mongolian script encoding in Unicode and Mr. Namujila's proposed solutions, as outlined in his proposal.

The following summarizes the discussion:

- We agree with several points made by Mr. Namujila: the current phonetic model is problematic, MVS/FVS and NNBSP have issues, the present model poses cybersecurity phishing risks, and there are multiple ways to encode words in Mongolian.
- The approach of Mr. Namujila is better termed "glyphic", rather than "graphetic." The "glyphic" approach is closer to the Arabic Presentation model (whereas "graphetic" is closer to the Arabic script model).
- The UTC has made the decision to improve the phonetic model, based on response from the standards bodies in China, Inner Mongolia of China, and Mongolia, the local community of Mongolian users, and various publishers, though Unicode had earlier voiced a preference for a graphetic approach. The goal of Unicode is to provide a stable solution for implementing Mongolian that has been reached by consensus from all the stakeholders.
- Separating Todo, Manchu, and Sibe as different scripts from the Mongolian script, which is suggested by Mr. Namujila, is in our opinion very problematic, because it will cause even more confusability than at present.
- The idea of a dual track – that is, the phonetic model and a new Mongolian block with a different model (such as the graphetic approach) – was raised earlier in Mongolian discussions but was eschewed in favor of an improved phonetic approach.

- More details are needed on the exact characters proposed by Mr. Namujila, how they were designed and the design principles behind them, as well as detailed analysis.
- News of Mongolia's decision to reintroduce the traditional Mongolian alphabet by 2025 and start a bi-scriptal practice will mean that reliable interchange of the script on computers and devices will be critical in the future.
- We encourage the Mongolian Working Group and its members to engage with Mr. Namujila.
- We recommend adding Liang Hai's slide set from IUC from IUC #42 from 2018 (https://lianghai.github.io/mongolian/whats-going-on-r1.pdf) to the set of Mongolian documents.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 7a of L2/20-105 (Script Ad Hoc Recommendations) to the author of MWG/4-N5 and MWG/4-N4.

7b. Mongolian proposal by Li Baowen
Document: MWG/4-N6 Suggestion about separately allocating code points for the "feminine and masculine characters" in the Mongolian encoding – Li Baowen

Comments: We reviewed this document which proposes disunifying eleven Mongolian characters for the Hudum, Todo, Manchu, Manchu Ali Gali, and Sibe writing systems. The overall goal of the author is to try to simplify shaping and get rid of gender-dependent shaping. He supports a new block for Mongolian, using a graphetic approach.

Recommendation: We recommend the UTC make the following disposition:
Assign an Action Item to Liang Hai to relay our appreciation to the author of MWG/4-N6 for the contribution and to offer to work with him on creating a revised document with actionable proposals.
Reference: Section 7b of L2/20-105 (Script Ad Hoc Recommendations).

7c. Hudum proposal by Ma Fuquan
Document: MWG/4-N7 A Proposal to Add Characters to Mongolian Coding – Ma Fuquan

Comments: We briefly reviewed this proposal to add characters to Unicode. In part one the author discusses loan words and proposes disunifying three Hudum characters for loan words. Part two of the document discusses compound words and proposes a new character for the tooth (*aleph*) at stem boundaries. Part three discusses NNBSP, recommending encoding an NNBSP dedicated to the Mongolian script. Part four discusses punctuation and proposed new characters dedicated to Mongolian script. Liang Hai can work with the author to explain the use of punctuation marks and relay that the Unicode Consortium is looking at different solutions for compounds and NNBSP.

Recommendation: We recommend the UTC make the following disposition:
Assign an Action Item to Liang Hai to relay our appreciation to the author of MWG/4-N7 for the

contribution and to offer to work with him on creating a revised document with actionable proposals.
Reference: Section 7c of L2/20-105 (Script Ad Hoc Recommendations).

7d. Todo proposal by Minzai
Document: MWG/4-N3 Proposal for a glyphic encoding for Todo – Minzai

Comments: We reviewed briefly this proposal for a glyphic encoding of Todo. The author proposes a new block for Todo, with a glyphic encoding (like Arabic Presentation Forms). The proposed glyphic model is problematic (see section 7a above).

Recommendation: We recommend the UTC make the following disposition:
Assign an Action Item to Liang Hai to relay our appreciation to the author of MWG/4-N3 for the contribution and to offer to work with him on creating a revised document with actionable proposals.
Reference: Section 7d of L2/20-105 (Script Ad Hoc Recommendations).

---

8. Tangsa
Document: L2/20-124 Preliminary Proposal to add the Tangsa Script in the SMP of the UCS – Morey

Comments: We reviewed this preliminary proposal for a script created in 1990 to write the Muishaung / Mossang variety of Tangsa spoken in Arunachal Pradesh, India. The Script Ad Hoc had seen an earlier version of this proposal, which builds off an introductory proposal by Anshuman Pandey (L2/13-231).

The following comments were raised during discussion:

- The script is an alphabet (i.e., having the tone on a vowel still qualifies it as an “alphabet” in terms of Unicode – the vowels are still distinct letters).
- The name “Tangsa” seems appropriate. If another Tangsa script is encoded, it could have a name that differentiates it from the script created by Mr. Lakhum Mossang.
- Change the names from TANGSA VOWEL to TANGSA LETTER.
- For characters whose names have tones, use a hyphen-digit format to indicate the tone number (i.e., “TONE-1”, “TONE-2”).
- How are the tones referred to by the creator and users (i.e., are there native names, rather than “TONE-1”, “TONE-3”, etc.)?
- Modify the vowel letter names so the tone attribute comes before vowel length and vowel quality, i.e., U+16A96 TANGSA LETTER TONE-1 UE, U+16A92 TANGSA LETTER TONE-1 SHORT UE.
- Change the name of U+16A91 TANGSA VOWEL UE LONG MID FALLING TONE to TANGSA LETTER TONE-2 LONG UE.
- Change the names for the syllabic nasals to “SYLLABIC M”: TANGSA LETTER TONE-1 SYLLABIC M, etc.
- In reference to “final position” (top of page 2), clarify whether “final position” refers to word final or syllabic final.
- The script still appears to be in development. We welcome further evidence of the script in use, such as in printed materials and its widespread acceptance by the community.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 8 of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-124.

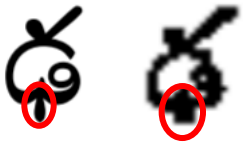
---

9. Telugu
9a. Telugu Sign Nukta
Documents:
L2/20-085 Revised Proposal to Encode Telugu Sign Nukta - Rajan et al.
Background docs:
L2/19-401 Proposal to Encode Telugu Sign Nukta –- Rajan et al.
L2/19-405 Additional Evidence for the use of Nukta sign in Telugu

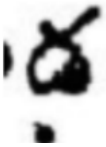
Comments: We reviewed this revised proposal for a single character, TELUGU SIGN NUKTA. The *nukta* is used in several Indic scripts to extend the character repertoire in order to represent non-native phonemes. Discussion at the January 2020 UTC meeting on the earlier Telugu *nukta* proposal, L2/19-401 (with additional evidence in L2/19-405), raised the question on the potential confusability of the *nukta* with the "teardrop" shape that appears in the glyphs for four Telugu letters (DDHA, CHA, PHA, and DHA). These four have corresponding non-aspirate letters with no teardrop in their glyphs (DDA, CA, PA, and DA).

Rajan has uncovered evidence that the *nukta* is productive, giving examples of the *nukta* appearing with six Telugu letters (GA, DDA, PHA, JA, KA and LLA), three more than identified in L2/19-401 and L2/19-405. In addition, Rajan demonstrates that the *nukta* is positioned very low in relation to the base letter (see below, center), quite different from teardrop, which is very close to the letter (below, left). The authors also provide evidence showing the circle glyph as a variant of the dot (below, right), also positioned very distant from the base letter.

TELUGU LETTER DDHA
(with teardrop circled in red)



TELUGU DDA with dot-nukta



TELUGU DDA with circle nukta

The following is a summary of the comments:

- The dot and circle appear to be variants of the *nukta*. No contrastive use of the dot-nukta and the circle *nukta* in the same document was presented. If evidence is found of a distinction between the dot and circle *nukta* in the future, they could be disunified.
- Although the dot *nukta* is more common, using the circle *nukta* as the representative glyph in the code chart will prevent possible confusion of the *nukta* dot and the teardrop. We recommend an appropriate annotation be added to the names list.
- The code point location appears to be good.

Recommendations: We recommend that the UTC approve the following:
SAH-UTC163-R5: The UTC accepts U+0C3C TELUGU SIGN NUKTA. Reference: L2/20-085.

We also recommend the UTC make the following disposition:
Assign an AI to the names list editor to review the proposed annotation for U+0C3C TELUGU SIGN NUKTA on page 4 of L2/20-085 and discussed in section 9a of L2/20-105 (Script Ad Hoc Recommendations).

9b. Telugu Letter Nakaara Pollu
Document: L2/20-084R Proposal to Encode Telugu Letter Nakaara Pollu – Rajan et al.

Comments: We reviewed this proposal to encode Telugu letter *nakaara pollu*, an historical vowel-less form of Telugu NA. The authors request the atomic encoding of the character, comparable to the atomic encoding of Bengali *khanda ta*. According to the authors, *nakaara pollu* is not supported by any fonts and atomically encoding the character would simplify rendering and representation.

The following points were raised during discussion:

- The current text in the block intro to Telugu (§12.7 of *TUS*) relies on a font-level solution. The text dates to Unicode 6.1 (2012), and was drawn from L2/11-409. *TUS* recommends the representation of *nakaara pollu* as a sequence <NA, VIRAMA> and, when prevented from conjunct formation with the following consonant, <NA, VIRAMA, ZWNJ, Consonant>.
- *Nakaara pollu* is the archaic use of a special ligature.
- The examples provided show contrastive use of *nakaara pollu* in the Old Style vs. New Style that appear in the same document.
- *Nakaara pollu* can be represented today, though it apparently not supported in fonts.
- Encoding a new character is a cleaner solution, in our opinion, though it will take time to be supported in fonts (once it is published in Unicode). However, it is an historic form that is relatively rare and poses no re-ordering issues.
- There was discussion on the spelling of NAKAARA in the character's name: should it be NAKARA or NAKAARA? Because there is no consistency in the spelling of long vowels in various Indic character names (cf. virama for Skt. *virāma*), the name NAKAARA was considered acceptable.

Recommendation: We recommend that the UTC approve the following:
SAH-UTC163-R6: The UTC accepts the following character for encoding in a future version of the standard: 0C5D TELUGU LETTER NAKAARA POLLU.
Reference: L2/20-084R.

9c. Telugu Block Intro Fix
Document: L2/20-086 Fixing a Superfluous Sequence in Telugu Section of *TUS* – Rajan

Comments: We reviewed this proposed fix in the "Rendering Behavior" section of the Telugu block intro (§12.7 of *TUS*, p. 501). The current text exemplifies a virama-attached letter with the sequence <KA,

VIRAMA, ZWNJ>. However, the ZWNJ is not needed in that context. We agree and recommend ZWNJ be removed from the first example in the "Rendering Behavior" section of the Telugu block intro (p. 501).

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the Editorial Committee to remove ZWNJ from the first example in the "Rendering Behavior" section of the Telugu block intro (p. 501) and to review all of §12.7 to ensure there is no other reference to the text with the error. Reference: Section 9c of L2/20-105 (Script Ad Hoc Recommendations).

9d. Telugu Reph
Document: L2/20-083 Proposal to Encode Telugu Reph – Rajan et al

Comments: We reviewed this request to encode Telugu *reph*, a character not used in modern Telugu, but found in historical and archaic orthographies. The authors propose the atomic encoding of the character in order to get better font-vendor support.

The following summarizes the points made during discussion:

- Currently, the reduced form of TELUGU RA (*reph*) is phonetically preposed in clusters, but the *reph* appears graphically on the right-hand side of a cluster. For example, the following is used to represent *rma:*
  U+0C30 ర *ra* + U+0C4D ◌్ *virama* + U+0C2E మ ma → మ్ర
  To represent *reph* explicitly in modern texts, ZWJ is employed after the virama. To prevent *reph* being displayed by default by a font for older texts, ZWJ is placed after RA but before the virama.
- Current text on *reph* in the Telugu section dates to Unicode 6.1
- In our view, the *reph* is already capable of being represented, though some fonts reportedly don't support it today. Encoding a new character to represent *reph* is not, in our opinion, the best answer, since fonts still may not support it. In addition, the failure of reordering during migration will lead to users hacking with the graphic encoding order and thus polluting texts.
- If problems continue, we recommend the authors re-submit a document showing the issues, verifying the *reph* works in Harfbuzz with a font, and provide additional rationale for encoding a new character.

Recommendation: We recommend the UTC make the following dispositions:
Assign an AI to the SAH to forward the comments in section 9d of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-083.

Assign an AI to the Editorial Committee to review the Telugu and Gurmukhi block intros for any text that refers to fonts. Reference: Section 9d of L2/20-105 (Script Ad Hoc Recommendations).

9e. Two Telugu Letters
Document: L2/20-119 Proposal to Encode Two Letters in Telugu – Rajan

Comments: We reviewed this proposal for two letters in Telugu, used to transcribe Tamil religious texts. The proposal includes ample evidence for the characters.

The proposal offers two options for representing LLA and RRA: (1) Use the Tamil letters LLLA and RRA (adding an entry in ScriptExtensions.txt). However, this approach will create problems for determining script runs. (2) Encode two new characters. We agree with the author that encoding two new characters would be preferable.

Both letters lack attested subjoined forms. We briefly discussed concerns about the RRA character, since its geminated form always appears with a virama on the first RRA (presumably mimicking the Tamil orthography), instead of having the second RRA taking a subjoined form. Should a ZWNJ be required to represent this spelling of <rra with virama, rra>? We welcome documents discussing the issue.

Recommendation: We recommend that the UTC approve the following:
SAH-UTC163-R7: The UTC accepts the following two characters for encoding in a future version of the standard:
0C5B TELUGU LETTER TAMIL TRANSCRIPTIONAL LLLA
0C5C TELUGU LETTER TAMIL TRANSCRIPTIONAL RRA
Reference: L2/20-119.

---

10. Wancho
Document: L2/20-121 Request to change code chart font – Scheuren

Comments: We reviewed this request to change the code chart for Wancho, which was published in Unicode version 12.0 in 2019. The author of the document, Zachary Scheuren, worked with the creator of the script, Banwang Losu, so the font being proposed for the code chart would better reflect what is being taught in schools.

The following comments were made:

- The scope of the changes in the proposed Wancho font is not as marked as those were for Adlam (L2/19-119R). Concerns would arise when the identity of a character is questioned because of glyphs re-design, but the changes in the Wancho glyphs do not appear to be radical.
- The author should provide additional evidence, clearly highlighting the new vs. old glyphs. Examples from books used in education strengthen the case for a font change. Filling out the proposal with details is especially useful; not much information is available on the script and this document can serve as a source. It can also become a model for those who want to change a script's chart font.
- Include information on the Wancho fonts currently available.
- Use a lighter weight font for figure 1.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the Editorial Committee to issue an erratum, based on L2/20-121. Reference: Section 10 of L2/20-105 (Script Ad Hoc Recommendations).

V. SOUTHEAST ASIA, INDONESIA, AND OCEANIA

11. Javanese

Reference: Action Item 143-A56a: Investigate feedback on Javanese from R.S. Wihananto in PRI #297 Unicode 8.0.0 Beta

Comments: The following summarizes R.S. Wihananto's comments from the Unicode 8.0 Beta (2015) on Javanese, shown in bold, and the Script Ad Hoc's responses. As detailed below, the comments on Javanese resulted in several actions in 2019, which need to be relayed back to R.S. Wihananto.

- **U+A9BD [JAVANESE CONSONANT SIGN KERET] in Indic Syllabic Category should be recategorized as 'Consonant_Medial' like U+A9BE [JAVANESE CONSONANT SIGN PENGKAL] and U+A9BF [JAVANESE CONSONANT SIGN CAKRA]. However, unlike U+A9BE [PENGKAL] and U+A9BF [CAKRA], U+A9BD [KERET] can't be followed by vowel signs because it already have inherent ĕ vowel. Also the Unicode Character Categories of this U+A9BD [KERET] character is incorrect. It should not be categorized as 'Mc' (Mark, Spacing Combining), but 'Mn' (Mark, Nonspacing). This character is nonspacing and its behavior in combining with other character and forming ligature is similar to nonspacing vowel sign u (U+A9B8) and uu (U+A9B9).**

  Comments from Script Ad Hoc:
  As noted by Norbert Lindenberg, the ISC of U+A9BD JAVANESE CONSONANT SIGN KERET was changed to ISC="Consonant_Medial" and gc=Mn in Unicode 12.0, based on L2/19-004. The consensus is recorded in L2/19-008.

- **The positional category of U+A9BE JAVANESE CONSONANT SIGN PENGKAL should be corrected from 'Right' to 'Bottom_And_Right'. The positional category of U+A9BF JAVANESE CONSONANT SIGN CAKRA should be corrected from 'Right' to 'Bottom_And_Left'; but I can't find this category in the Indic Positional Category data. This character is similar to U+103C MYANMAR CONSONANT SIGN MEDIAL RA. U+103C is not found/categorized in the Indic Positional Category data.**

  Comments from Script Ad Hoc:
  As noted by Norbert Lindenberg, the positional category of U+A9BE JAVANESE CONSONANT SIGN PENGKAL was changed to "Bottom_and_Right" in Unicode 13.0, based on L2/19-083. The consensus is recorded in L2/19-122. The positional category *o*f U+A9BF JAVANESE CONSONANT SIGN CAKRA has been Bottom_and_Left since Unicode. 10.0.

Recommendation: We recommend the UTC make the following disposition:
The UTC forwards the comments in section 11 of L2/20-105 (Script Ad Hoc Recommendations) to R.S. Wihananto.

12. Western Cham

Documents: L2/20-061R Final Proposal to encode Western Cham in the UCS – Hosken
L2/20-122 Comments on L2/20-061R Final Proposal to encode Western Cham in the UCS – Anderson

Related documents:
L2/20-063 Response to Western Cham Script Adhoc Comments L2/20-046 – Hosken
L2/20-018 Response to proposal to encode Western Cham in the UCS (L2/19-217) – Alberto Pérez Pereiro, et al.
L2/20-062 Response to L2/20-018 W. Cham response (document above from Pérez Pereiro) – Hosken

Comments: We reviewed the L2/20-122 document, since it highlighted changes in the latest version of the Western Cham proposal, provided the responses to the earlier Script Ad Hoc recommendations (if there were any), and gave additional comments, all in one document. It also included a version of the Western Cham proposal with significant changes highlighted, so it would be easier to see the modifications. The goal of the comments is ultimately to help the author improve the proposal, and to end up with a proposal that serves the entire user community.

The following comments were raised during discussion:

- Because some comments or questions were not answered, and, in some cases the information was not incorporated in the revised proposal, we recommend the L2/20-122 document be sent to the author of the Western Cham proposal.
- We believe it would help UTC members to make a more informed decision if there were a brief discussion of choice between logical and visual orders. Hence, it is recommended that this be added.
- The fact that the proposal does not acknowledge the printed materials in L2/20-018 (some of which were in the earlier Western Cham proposal (L2/16-198) remains problematical. We encourage the two groups represented by the proponents of the proposal and L2/20-018 to collaborate.

Note: See also section ***18.*** in the Public Review Feedback at the end of this document for feedback on Western Cham.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 12 of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-061R.

VI. EAST ASIA

13. Nushu

Document: L2/20-075 Updates for 8 values for 4 Nushu characters in UCD - EIso Chan

Comments: We reviewed this request to correct 8 values for Nüshu characters in the NushuSources.txt file. The proposed corrections appear to be valid.

Recommendation: We recommend the UTC make the following dispositions:
Assign an AI to the Editorial Committee to issue an erratum, based on L2/20-075. Reference: Section 13 of L2/20-105 (Script Ad Hoc Recommendations).

VII. SYMBOLS AND NUMERICAL NOTATION SYSTEMS, AND OTHER SCRIPT TOPICS

14. Kaktovik Numerals

L2/20-070 Exploratory proposal to encode the Kaktovik numerals – Marín Silva

Comments: We reviewed this exploratory proposal for the Kaktovik numerals, a vigesimal (base 20) number system. The numerals were initially created as part of a middle school activity on Barter Island in northern Alaska by students whose language is Iñupiaq, a language that uses base 20. The Kaktovik number system was taught in the mid-1990s in northern Alaska and has spread from there to other areas. It was adopted in 1996 by the Commission on Inuit History Language and Culture.

The following summarizes the comments:

- The proposal includes charts, but only one example of the numbers in use (figure 7, with link to YouTube video). Provide examples in print showing how a date and fractions would be written, as well as examples of the numbers being used in common problems in arithmetic.
- The proposer should check with the user community on the character names and provide an update on the usage of the numerals in schools today.
- We recommend the following two columns be used: U+1D250..U+1D26F.

Recommendation: We recommend the UTC make the following dispositions:
The UTC forwards the comments in section 14 of L2/20-105 (Script Ad Hoc Recommendations) to the author of L2/20-070.

Assign an AI to the Roadmap Committee to allocate U+1D250..U+1D26F for Kaktovik Numbers. Reference: Section 14 of L2/20-105 (Script Ad Hoc Recommendations) and L2/20-070.

---

15. Quikscript

L2/20-090 Comparison of Shavian vs. Quikscript – Anderson
Reference doc: Codechart for Shavian and Shavian Quikscript Extensions (2007) – Everson

Comments: We discussed a November 2019 request to ISO 15924 from John Cowan for a script variant code of Shaw (Shavian) for Quikscript. The request was relayed to the Script Ad Hoc for discussion. We also briefly examined a comparison of the repertoire of Shavian vs. Quikscript.

In his later correspondence with Markus Scherer, John Cowan writes:

> Although the Shavian and Quikscript scripts are very similar, they are only somewhat mutually legible when used to write English (which is the major use, although it is possible to write Welsh and Scottish Gaelic with Quikscript). Many letters are the same, but others are used only in one

script and a few have entirely different meanings in the two scripts. This is comparable to Latn vs Latg: the Latg "s" looks to someone who only knows Latn like an "r", and the Latg "G/g" are completely illegible.

I would of course have no objection to a completely separate script code, but it's not like there is no mutual legibility at all. The two scripts are definitely more similar than Latn, Cyrl, and Grek, for example, where mutual legibility is basically zero**.**

The following summarizes comments:

- Could Quikscript be handled by Shavian code points, but with a change of font, or is it a completely separate script? Or should it be handled as an extension of Shavian?
- A document from Michael Everson in 2007, which is linked to "ShavianQS" on the SMP Roadmap, provides no context, only charts of Shavian and Quikscript. The currently encoded Shavian repertoire is on the left of the first page of the 2007 document and a chart of Shavian with added characters and glyphs for Quikscript on the right-hand side of page 1. A third chart entitled "Shavian Quikscript Extensions" includes just Quikscript characters and glyphs (on page 2). Note: The original Shavian proposal L2/97-103 (revised L2/01-256) did not discuss Quikscript.
- In order to answer the question about the status of Quikscript, more information is needed. A document discussing the differences between Shavian and Quikscript and the options is required.

Recommendation: We recommend the UTC make the following dispositions:
The UTC forwards the comments in section 15 of L2/20-105 (Script Ad Hoc Recommendations) to John Cowan.

---

16. Unicode Symbols representing Disasters
Document: L2/20-078 LS [Liaison Statement] on Unicode symbol numbers representing disasters – 3GPP TSG CT WG 1

Comments: We reviewed this request from a technical specification group (TSG) from 3rd Generation Partnership Project (3GPP), a standards organization that develops protocols for mobile telephony. Working Group 1, which submitted the request, was from Core Network and Terminals group (CT). WG1 is currently working on a Public Warning System for 5G and wants to improve text-based warning messages for those who are not literate in the language of the text being sent.

The document asked SC2 to provide Unicode symbols conveying various disasters (earthquake, tsunami, fire, and 8 other disasters) so they can be included in public warning messages. If such symbols are not present, the WG requests they be standardized.

In our opinion, SC2 and the Unicode Technical Committee are not in the business of designing icons for concepts. WG1 is welcome to select whichever symbols they consider appropriate from ISO/IEC 10646, but we offer no recommendations. WG1 is invited to propose characters for inclusion in ISO/IEC 10646, but note that SC2 and the Unicode Technical Committee don't assign code points to each and every iconic design that potentially could be considered a pictographic character. To encode new characters, inline usage as a symbol in what could be considered plain text is required. For example, simple use as

graphic icons, placard style, printed on labels, such as dry-cleaning advisory icons, etc., may not suffice to indicate appropriate use as text characters. Useful guidelines on proposing new characters are described at: http://www.unicode.org/pending/proposals.html

Recommendation: We recommend the UTC make the following dispositions:
The UTC forwards the comments in section 16 of L2/20-105 (Script Ad Hoc Recommendations) to the submitters of L2/20-078.

VIII. PUBLIC REVIEW FEEDBACK ON SCRIPTS (L2/20-104)
Topics: Gurmukhi, Western Cham, Legacy Computing, Old Chinese Iteration Marks, Devanagari

17. Gurmukhi (from Public Review feedback L2/20-104)
Date/Time: Tue Jan 28 18:57:28 CST 2020
Name: Sarabveer Singh
Report Type: Feedback on an Encoding Proposal
Opt Subject: Suggestions for Gurmukhi Bindi Before Bihari (L2/18-319, L2/19-167, L2/19-283)

> Singh in L2/18-319 and L2/19-167 wishes for the Unicode specification to add support for GURMUKHI SIGN BINDI and GURMUKHI TIPPI to display before GURMUKHI VOWEL SIGN II. As noted in L2/19-047, this combination is most likely to be a stylistic difference.
>
> However, this combination should be supported as a stylistic option in Unicode fonts. In testing, I have only found the "liga" and/or the "rlig" OpenType lookups display this stylistic combination. This is an unsupported method and does not work universally on different systems. In my experience, the ligature displays correctly in the major web browsers (Google Chrome, Mozilla Firefox, Apple Safari), but they do not display correctly in Microsoft's software (Office, Edge, Internet Explorer).
>
> I request that a OpenType Ligature Lookup Table be recommend to implement this stylistic combination in Unicode fonts, such as the "abvf" Lookup Table.

Comments: We reviewed this comment on Gurmukhi BINDI before BIHARI. While Sarabveer Singh believes BINDI and TIPPI to be stylistic (echoed in his document L2/20-060), it is not yet clear whether the display of BINDI and TIPPI before GURMUKHI VOWEL SIGN II is stylistic or plain text. If it is deemed stylistic, then the author would be welcome to contact the OpenType community to discuss the appropriate implementation.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 17 of L2/20-105 (Script Ad Hoc Recommendations) to Sarabveer Singh.

---

18. Western Cham (from Public Review feedback L2/20-104)
Date/Time: Sat Feb 1 17:56:25 CST 2020

Name: Doug Ewell
Report Type: Feedback on an Encoding Proposal
Opt Subject: Comments on L2/20-061, Final Proposal to encode Western Cham in the UCS

> L2/20-061 proposes, among other characters, a group of eight characters for Western Cham lunar month names (ARABIC SYMBOL ONE DOT LUNAR MONTH through ARABIC SYMBOL SEVEN DOTS LUNAR MONTH), to be placed in the Arabic Mathematical Alphabetic Symbols block at code points U+1EEF8 through U+1EEFF.
>
> The Arabic Mathematical Alphabetic Symbols block was intended for stylistic variations of existing Arabic letters, to be used in special mathematical contexts. It is analogous to the Mathematical Alphanumeric Symbols block for existing Latin and Greek letters and digits. It is not intended for encoding of new "normal" characters. The proposed characters are "special" in that they are used only in Western Cham and only for lunar month names, but they are not "mathematical"; they are not used to represent variables, constants, sets, etc. in mathematical expressions.
>
> Both the text and the proposed Unicode properties show that the proposed characters are not stylistic variations of existing Arabic letters, and do not follow the pattern of other characters in this block:
>
> 1EEF8;ARABIC SYMBOL ONE DOT LUNAR MONTH;So;0;ON;;;;;N;;;;;
> cf.
> 1EE00;ARABIC MATHEMATICAL ALEF;Lo;0;AL;<font> 0627;;;;N;;;;;
>
> They are "symbols" (So), not "letters" (AL), and are not <font> varieties of existing letters.
>
> In the revision history, it was noted that these characters were moved in Revision 3 (November 2019) from the proposed Western Cham block to this block. Item 6 in the section "Repertoire" includes an inadvertent lingering reference to ARABIC SYMBOL SEVEN DOTS LUNAR MONTH being encoded at U+1E26F.
>
> I recommend moving these eight symbols back into the proposed Western Cham block, as they were before Revision 3. I have no objection at all to encoding these symbols, only to this particular proposed location.

Comments: We reviewed this feedback on the proposed Western Cham lunar month characters, which are currently located in the Mathematical Alphanumeric Symbols block. We agree that the location of the eight lunar month may be better suited for another location, since the characters are not specific presentation forms of existing Arabic letters. Other possible locations could be in the Western Cham block, in the Arabic Presentation Forms blocks, or in a new block.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 18 of L2/20-105 (Script Ad Hoc Recommendations) to Doug Ewell and Martin Hosken.

19. Legacy Computing (from Public Review feedback L2/20-104)
Date/Time: Sun Feb 16 15:29:46 CST 2020
Name: Arnim Sauerbier
Report Type: Feedback on an Encoding Proposal
Opt Subject: Symbols for Legacy Computing 1FB3C...

> The current proposal for "Symbols for Legacy Computing" unfortunatly lacks two-character full-triangles.
>
> Recreating the 'square font' diagonal triangles for legacy computing on modern systems require double-width triangles. The extant proposal only allows creation of triple-width triangles, forming a non-square aspect ratio.
>
> The following additions would allow drawing roughly 1:1 bilateral triangles out of two adjacent characters.
>
> A two-glyph triangle on bottom left, consisting of:
> LOWER LEFT BLOCK DIAGONAL CENTER LEFT TO LOWER RIGHT
> LOWER LEFT BLOCK DIAGONAL UPPER LEFT TO CENTER RIGHT
>
> A two-glyph triangle on bottom right, consisting of:
> LOWER RIGHT BLOCK DIAGONAL BOTTOM LEFT TO CENTER RIGHT
> LOWER RIGHT BLOCK DIAGONAL CENTER LEFT TO UPPER RIGHT
>
> A two-glyph triangle on upper left, consisting of:
> UPPER LEFT BLOCK DIAGONAL LOWER LEFT TO CENTER RIGHT
> UPPER LEFT BLOCK DIAGONAL CENTER LEFT TO UPPER RIGHT
>
> A two-glyph triangle on upper right, consisting of:
> UPPER RIGHT BLOCK DIAGONAL UPPER LEFT TO CENTER RIGHT
> UPPER RIGHT BLOCK DIAGONAL CENTER LEFT TO LOWER RIGHT
>
> These new codepoints are necessary to recreate Legacy Computing Graphics.
>
> I have been using PETSCII and other legacy-computing graphics since 1981, and am happy to answer any questions you may have.
>
> Thank you for your consideration in preserving legacy computer art.
>
> Arnim

Note: An Error Report was also received on Feb. 16, 2020 from Jean Larapiere requesting the same characters.

Comments: We reviewed the comments on Legacy Computing, which requested eight additional characters to create 'square font' diagonal triangles, apparently the left and right halves of U+25E2 through U+25E5.

Lisa Moore contacted two experts, Doug Ewell and Rebecca Bettencourt, who replied they weren't aware the characters were encoded in any characters sets (including PETSCII), though they acknowledge they are present in the UNSCII font. In order to evaluate the request, evidence should be provided, with specific information on the systems that use the characters.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 19 of L2/20-105 (Script Ad Hoc Recommendations) to Arnim Sauerbier and Jean Larapiere.

---

20. Old Chinese Iteration Mark (from Public Review feedback L2/20-104)
Date/Time: Thu Feb 20 12:45:44 CST 2020
Name: Markus Scherer
Report Type: Error Report
Opt Subject: review Script of U+16FE3 OLD CHINESE ITERATION MARK

> For consideration by script ad hoc & UTC
> Unicode 13 adds U+16FF0/1 Vietnamese reading marks with sc=Hani (and gc=Mc).
> In the same block is U+16FE3 OLD CHINESE ITERATION MARK sc=Zyyy (and gc=Lm).
>
> In discussion, Ken W. said that this one "patterns similarly to the modern iteration mark 3005. That one *is* sc=Han." And "So it would seem reasonable to me (in a *future* version) to ask for 16FE3 to change to sc=Han"
>
> Please review.

Comments: We reviewed this request to consider changing the script property value for U+16FE3 OLD CHINESE ITERATION MARK from Common (Zyyy) to Han. Many CJK punctuation marks in Scripts.txt are Common, which makes sense because the punctuation marks are used across several scripts. The Old Chinese iteration character, however, appears only in Old Han sources, and is not carried into any other script. Hence, in our view, the request to change the script property value to "Han" is justified.

A similar case was pointed out by Roozbeh Pournader for U+16FE2 OLD CHINESE HOOK MARK, which should also likely be changed to sc=Han.

Recommendation*:* We recommend that the UTC approve the following:
SAH-UTC163-R8: The UTC make a change in the Script property value for U+16FE3 OLD CHINESE ITERATION MARK from sc=Zyyy (Common) to sc=Han in Scripts.txt. Reference: L2/20-104.

SAH-UTC163-R9: The UTC make a change in the Script property value for U+16FE2 OLD CHINESE HOOK MARK from sc=Zyyy (Common) to sc=Han in Scripts.txt. Reference: L2/20-104.

---

21. Devanagari (from Public Review feedback L2/20-104)
Date/Time: Wed Apr 8 00:37:28 CDT 2020
Name: Barun Kumar Sahu
Report Type: Error Report
Opt Subject: "Devanagari sign avagraha" followed by "Devanagari sign anusvara" or "Devanagari sign candrabindu"

> Ideally, "Devanagari sign avagraha" (U+093D) can be followed by "Devanagari sign anusvara" (U+0902) or "Devanagari sign candrabindu" (U+0901). However, some word-processors do not accept this combination.
>
> My question is: Can "Devanagari sign avagraha" be followed by "Devanagari sign anusvara" or "Devanagari sign candrabindu" as per the Unicode Standard?
> (I think it should be allowed. For example, we should be able to write कऽंप or कऽॅप.)

Comments: We reviewed this feedback asking whether DEVANAGARI SIGN AVAGRAHA can be followed by DEVANAGARI SIGN ANUSVARA or DEVANAGARI SIGN CANDRABINDU.

The co-occurrence of *avagraha* with a following *anusvara* or *candrabindu* is not normally found in Sanskrit text. The combination appears to work in Harfbuzz, but apparently not in Windows 10 or the recent Mac OS. Can the feedback submitter provide examples of such combinations in any books? Liang Hai reports the topic was recently taken up in a discussion about TeX, and he can investigate other use cases, in consultation with the feedback submitter.

Recommendation: We recommend the UTC make the following disposition:
Assign an AI to the SAH to forward the comments in section 21 of L2/20-105 (Script Ad Hoc Recommendations) to Barun Kumar Sahu.