

Repha representation for Kawi

Norbert Lindenberg

2020-12-06

This document discusses the options for representing the repha in the Kawi script, which is proposed for encoding in [L2/20-284 Proposal to encode Kawi in the UCS](#). The option chosen is to encode the repha as a separate character with the general category Lo and the Indic syllabic category Consonant_Preceding_Repha, which implies placing this character immediately before the base consonant of the cluster.

Repha

In Unicode parlance, *repha* is a special presentation form of a cluster-initial dead consonant *ra*, which is rendered as a mark above or to the right of a subsequent consonant that serves as the base of the cluster. An extension of the repha concept is the Myanmar kinzi, which applies the same idea to a cluster-initial dead consonant *nga* for the Burmese language, and to some other consonants in other languages written in the script. In the Balinese, Javanese, and Kawi scripts, marks that originally were a repha have also been used as final *-r*. In these three scripts, the repha or final *-r* is usually positioned above the base consonant.

Repha representations for already encoded scripts

The Unicode Standard has used a bewildering variety of approaches to representing repha and repha-like marks:

1. An initial *ra* followed by a dependent vocalic liquid may be rendered as repha above the corresponding independent vocalic liquid, or in the original form, under conditions not specified in the Unicode Standard. This approach is used for the Bhaiksuki, Devanagari, Gujarati, Kannada, Oriya, and Telugu scripts, although it is only mentioned in the specification for Devanagari.
2. An initial *ra* followed by a virama not followed by ZWJ represents repha, which needs to be reordered so that it applies to the base of the cluster. The presence of ZWJ before the virama indicates that the initial *ra* remains the cluster base. The presence of ZWJ after the virama indicates that the initial *ra* is rendered in eyelash form. This approach is used for the Devanagari script.
3. An initial *ra* followed by a virama not followed by ZWJ represents repha, which needs to be reordered so that it applies to the base of the cluster. The presence of ZWJ after the virama indicates that the initial *ra* is rendered in eyelash form. This approach is used for the Modi script.
4. An initial *ra* followed by a virama followed by ZWJ represents repha, which needs to be reordered so that it applies to the base of the cluster. The absence of ZWJ after the virama indicates that the initial *ra* is rendered in eyelash form. This approach is used for the Newa script.

5. An initial *ra* followed by a virama represents repha, which needs to be reordered so that it applies to the base of the cluster. The presence of ZWJ before the virama indicates that the initial *ra* remains the cluster base. This approach is used for the Bengali and Kannada scripts.
6. An initial *ra* followed by a virama represents repha, which needs to be reordered so that it applies to the base of the cluster. This approach is used for the Gujarati and Oriya scripts.
7. An initial *ra* followed by a virama followed by ZWJ represents repha, which needs to be reordered so that it appears to the right of the the base of the cluster. The absence of ZWJ indicates that the initial *ra* remains the cluster base. This approach is used for the Gurmukhi script.
8. An initial *ra* followed by a virama followed by ZWJ represents repha, which needs to be reordered so that it appears to the right of the the base of the cluster. The absence of ZWJ, or the presence of ZWJ before the virama, indicates that the initial *ra* remains the cluster base. This approach is used for the Telugu script if a repha is needed in modern texts.
9. An initial *ra* followed by a virama represents repha, which needs to be reordered so that it appears to the right of the base of the cluster. The presence of ZWJ before the virama indicates that the initial *ra* remains the cluster base. This approach is used for the Telugu script if a repha is needed or not needed in older texts. The Unicode Standard does not specify how implementations would distinguish between modern and older texts.
10. An initial *ra*, *nga*, or Mon *nga* followed by a pure killer followed by an invisible stacker represents a kinzi, which needs to be reordered so that it applies to the base of the cluster. This approach is used for the Myanmar script.
11. The repha is encoded as a letter with Indic syllabic category Consonant_Preceding_Repha, which must be placed before the base of the cluster, but for rendering needs to be converted into a combining mark and reordered so that it applies to the base of the cluster. This approach is used for the Dives Akuru, Malayalam, and Masaram Gondi scripts.
12. The repha is encoded as a unique code point with Indic syllabic category Consonant_Succeeding_Repha, which must be placed after the base of the cluster. This approach is used for the Khmer script.
13. The repha is encoded as a combining mark with Indic syllabic category Consonant_Final, which must be placed at the end of the syllable. This approach has been adopted for the Balinese and Javanese scripts starting from Unicode 14, because in these scripts repha is an exceptional and older use of a character that today is a final consonant. The encoding records the mark, not whether it's used as an initial or final consonant.
14. The encoding of repha is unspecified. This approach has been adopted at least for the Grantha script, even though the proposals by [Naga Ganesan](#) and [Shriramana Sharma](#) provide documentation on repha with different suggested encodings.

The following table shows the repha representations for those scripts where the Unicode Standard or Unicode data specify them, contrasted with any specified representations of a cluster-initial dead consonant *ra* in nominal form or of *eyelash ra*, another special presentation form. Characters forming the context of repha are shown in gray.

Script	Nominal <i>ra</i> glyph for initial dead <i>ra</i>	repha	eyelash <i>ra</i>
Balinese	ra virama consonant	consonant vowel? -r	
Bengali	ra ZWJ virama consonant	ra virama consonant	
Devanagari	ra ZWJ virama consonant	ra virama consonant	ra virama ZWJ consonant
Dives Akuru		repha consonant	
Gujarati		ra virama consonant	
Gurmukhi	ra virama consonant	ra virama ZWJ consonant	
Javanese	ra virama consonant	consonant vowel? -r	
Kannada	ra ZWJ virama consonant	ra virama consonant	
Khmer	ra stacker consonant	consonant repha vowel?	
Malayalam	ra virama consonant	repha consonant	
Masaram Gondi	ra virama consonant	repha consonant	
Modi		ra virama consonant	ra virama ZWJ consonant
Myanmar	ra stacker consonant	[nga ra Mon-nga] killer stacker consonant	
Newa		ra virama ZWJ consonant	ra virama consonant
Oriya		ra virama consonant	
Telugu (modern)	ra virama consonant, ra ZWJ virama consonant	ra virama ZWJ consonant	
Telugu (older)	ra ZWJ virama consonant	ra virama consonant	

Repha representation for Kawi

According to the [Proposal to encode Kawi](#), a cluster-initial dead consonant *ra* usually takes a repha form; however, in some cases it also remains in its nominal form. In addition, the repha form is also occasionally used for the final consonant *-r*.

This compares to the Balinese and Javanese scripts (which both have evolved from Kawi) as follows:

- In Kawi, a cluster-initial dead consonant *ra* usually takes a repha form. In Balinese and Javanese it usually remains in the nominal form.

- In Kawi, the repha glyph usually is a repha; only in late writing is it sometimes used as a final consonant *-r*. In Balinese and Javanese, the corresponding glyphs are usually used as final *-r*; only in early writing are they sometimes used as repha.

Which of the existing repha representations could be used for Kawi? Here are some that should be excluded:

- Leaving the representation unspecified would likely result in incompatible implementations. This excludes approaches 1, 14, and the pair 8 and 9.
- Representations that provide no distinction between a nominal *ra* glyph and a repha glyph can't be used because Kawi needs that distinction. This excludes approaches 3, 4, and 6.
- Representations that rely on the use of ZWJ for distinctions have fallen out of favor, as such control characters often cause problems in text input and processing. This excludes approaches 2, 5, 7, 8, and 9.
- A representation that requires both a killer and a stacker to follow a consonant is quite ugly. It was introduced for Myanmar to replace an earlier [representation with ZWJ](#), and has enabled the representation of kinzi forms for at least three, possibly more, consonants in this script. However, the latter is not needed for Kawi, which has only one repha. This excludes approach 10.
- Representations that encode repha and final *-r* separately would be beneficial for text operations such as string comparison, but would be unreliable because users can't see the difference and therefore may input the wrong one.

This leaves us with the representation of the repha glyph, used as either repha or final *-r*, as a single separate character. The options for Indic syllabic category and placement are:

- `Consonant_Preceding_Repha`, placement before the base consonant. If the character is used as a repha (the normal case), this placement corresponds to the pronunciation and simplifies string comparison for sorting. If the character is used as final *-r*, string comparison for sorting has to move it across the entire cluster to the end. The general category used for all current `Consonant_Preceding_Repha` characters is `Lo`, which creates the risk of having the character separated from the base it belongs to. For line breaking, the repha character would need category `BB` to prevent separation. Extended grapheme cluster formation has special case handling for `Consonant_Preceding_Repha`. For rendering, the character has to be moved to after the base; the Universal Shaping Engine description specifies how to support this, but then warns that this is not currently supported. Testing with Masaram Gondi showed that it is supported in HarfBuzz, but not yet in CoreText and DirectWrite. As the category is already used by two other scripts handled by the Universal Shaping Engine, Masaram Gondi and Dives Akuru, there is a reasonable chance that these implementations will be fixed. Keyboards would have to move this character, which users are likely to input after the base, to its placement before the base.
- `Consonant_Final`, placement at the end of the syllable. If the character is used as final *-r*, this placement corresponds to the pronunciation and simplifies string comparison for sorting. If the character is used as repha (the normal case), string comparison for sorting has to move it across the entire cluster to the front. `Consonant_Final` would match the corresponding characters in Balinese and Javanese; however, while use as final *-r* is the normal case in these modern scripts, it is an

exception in Kawi. Text segmentation is unproblematic because the mark follows the base. For text rendering, see the next section.

- Consonant_Succeeding_Repha, placement after the base consonant or possibly after a sub- or postjoined consonant. This placement doesn't correspond to the pronunciation of either usage, but sits in the middle between the two above. String comparison for sorting always has to move the character across part of the cluster. Text segmentation is unproblematic because the mark follows the base. However, there's no precedent for correct implementation of Consonant_Succeeding_Repha – the use of this category for Balinese, Javanese, and Sundanese had to be [corrected for Unicode 14](#), and specifications and implementations of Khmer [diverge in their handling of robat](#) in all but the simplest cases. In particular, it's not clear whether the repha character would, in the presence of a sub- or postjoined consonant, be placed right after the base consonant or after the sub- or postjoined consonant. The comment in the Unicode data file IndicSyllabicCategory.txt only says "when succeeding the main consonant"; the Khmer section of the Unicode Standard allows either position; the Universal Shaping Engine (which currently doesn't support Khmer, but supports Balinese, Javanese, and Sundanese based on Unicode 13 data) places it at the end of the syllable. From a rendering point of view, placement right after the base consonant would work better for Kawi, because the repha is usually attached to the base consonant, not any postjoined consonant. For Khmer, I don't know where the robat would attach in such a case, as it is not used together with sub- or postjoined consonants in modern Khmer, and I don't have sufficient information about middle Khmer.

The option chosen is to classify the repha as Consonant_Preceding_Repha, with placement right before the base consonant. This puts the character in the phonetically correct position in the syllable when it's used as repha, and aligns with the encoding of repha in the Masaram Gondi and Dives Akuru scripts. The implementations of the Universal Shaping Engine that don't fully support Consonant_Preceding_Repha yet will need to be updated to do so.

Acknowledgements

I'd like to thank Aditya Bayu Perdana and Ilham Nurwansah for their research into the use of conjunct forms for the consonant *ra* in Kawi, and the Unicode Script Ad Hoc for feedback on an earlier version of this document.