Proposal on Clarification and Consolidation of the Function of ZERO WIDTH JOINER in Indic Scripts

Date:	*2004-06-30*
Author:	Peter Constable, Microsoft
Address:	One Microsoft Way Redmond, WA 98052 USA
Tel:	+1 425 722 1867
Email:	petercon@microsoft.com

1 Introduction

The representational model for encoding Indic scripts in Unicode is described in Chapter 9 of the Standard, with sections describing details for each of the individual scripts. The first section, covering the Devanagari script, provides a much greater level of detail than do subsequent sections, and is provided as a template on which other scripts are based:

> The principles of the Indic scripts are covered in some detail in this introduction to the Devanagari script. The remaining introductions to the Indic scripts are abbreviated but highlight any differences from Devanagari where appropriate. *[Unicode 4.0, p. 221.]*

The implication, then, is that encoding formalisms specified in section 9.1 are considered applicable to all Indic scripts unless specified otherwise in subsequent sections for any particular script.

One problem with this arrangement is that the Indic scripts are not all the same; in fact, there are some very significant differences between scripts. Two particular problems resulting from differences are that it is not clear how certain encoding formalisms specified in section 9.1 are to be applied in other Indic scripts, and that there are common problems found in other scripts that are not addressed in the section on Devanagari.

The intent of this proposal is to rectify these problems, clarifying how the ZERO WIDTH JOINER (ZWJ) is to be applied in scripts that are unlike Devanagari, and consolidating common mechanisms for equivalent problems that exist in several scripts other than Devanagari.

The scope for what is proposed covers the nine scripts of India: Devanagari, Bengali, Gurmukhi, Gujarati, Oriya, Tamil, Telugu, Kannada and Malayalam.[1]

[1] It is intended that similar specifications should be applied as appropriate to Sinhala as well, though because that script has some significant differences it cannot simply be grouped together with the nine scripts identified here. (See note 10 for further details.)

2 Background

2.1 Dead consonants and conjoining forms in Devanagari

The function of ZWJ specified in Section 9.1 relates to the formation of halant and conjoining forms. These are well-known characteristic of Indic scripts. Consonants have inherent vowels associated with them, and therefore a vowel-less, or "dead", consonant generally requires special marking to indicate the lack of vowel.[2] For a syllable-final consonant, this can be shown using a mark known as "halant" (or "hasant", "hoshonto", etc. depending on the language). In encoded text, a dead consonant is represented as a character sequence consisting of the given consonant followed by a VIRAMA.

Devanagari character sequence	Halant form
< KA क, VIRAMA ् >	क्

Table 1. Devanagari conjunct ligature

When a syllable contains an initial consonant cluster, all but the last consonant in the cluster is dead. In encoded text, each dead consonant is represented as the given consonant followed by a VIRAMA. In terms of visual representation, the presence of dead consonants within a consonant cluster is very often reflected in that some of the consonant symbols take on variant forms or the symbols merge together into ligature "conjunct" forms.

Indic conjoining forms are very often explained in terms of the conjoining forms of Devanagari. In broad terms, Devanagari conjoining forms are of two types: conjunct ligatures, and non ligature conjoining forms known as "half" forms.

Devanagari, like most Indic scripts, has some conjunct ligature forms that are used for particular consonant clusters. A familiar example is the conjunct ligature for /kssa/:

Devanagari character sequence	Conjunct ligature
< KA क, VIRAMA ् , SSA ष >	क्ष

Table 2. Devanagari conjunct ligature

The Devanagari non-ligature conjoining forms are known as "half" forms since the variant conjoining form is generally derived by removing part of the nominal form. Typically (though not always), the portion that is removed is a vertical stem that can be perceived as a realization of the inherent vowel /a/. A half form of a consonant often connects with the following consonant, as is shown in Table 3 using the consonant KA:

[2] There may be cases in which a phonologically-dead consonant is not distinguished in written form from the "live" consonant with its inherent vowel. In Bengali, for instance, phonological changes have taken place over time while spellings have not changed, with the result that phonologically-dead consonants are often written the same as live consonants.

Consonant	Full form	Half form	Half form with following consonant
Devanagari GA	ग	ग्	ग्क
Devanagari PA	प	प्	प्क
Devanagari KA	क	क्	क्क

Table 3. Devanagari full vs. half forms

Devanagari also has a special-case conjoining form known as "reph": a dead RA at the start of a consonant cluster is realized as a mark over the main (full-form) consonant element of the cluster:[3]

Devanagari character sequence	Conjoining reph
< RA र, VIRAMA ◌्, KA क>	र्क

Table 4. Devanagari conjoining reph forms

2.2 Function of ZWJ in Devanagari

For most Indic scripts, there have been variations in typographic conventions that have been used, particular in relation to the presentation of consonant clusters. More specifically, a given consonant cluster may be written in some documents as a conjunct ligature, but in other documents as a non-ligature conjoining form.

One impact of this is that is that Unicode does not in general require conforming implementation to display consonant clusters in their conjoined form: for a given consonant cluster sequence < C1, VIRAMA, C2 >, if a font supports a conjunct ligature, it should display that form; but if a font does not support the ligature, it can display a non-ligature conjoining form instead; and if it has neither of those forms, then it can display an overt-halant form.

This variation has also resulted, however, in a need for some way to control the visual presentation for given-consonant cluster character sequence, so that an author could exert some measure of control over which form would be used to present a cluster.

In Devanagari, the typographic variation would entail the difference between creating a conjunct ligature versus using a half form. The mechanism defined in Section 9.1 for explicit control to request a half form is to insert the ZWJ after the VIRAMA:

[3] Devanagari script has two other special-case conjoining forms known as eyelash ra and vattu. It is not necessary to explain these at this point in order to capture the generalities.

Devanagari character sequence	Presentation
< KA क, VIRAMA ◌्, SSA ष >	क्ष
< KA क, VIRAMA ◌्, **ZWJ**, SSA ष >	क्‍ष

Table 5. ZWJ used to explicitly request a half form

As explained in Section 9.1,

> In certain cases, it is desirable to prevent a dead consonant from assuming full conjunct formation yet still not appear with an explicit virama. In these cases, the half-form of the consonant is used. To explicitly encode a half-consonant form, the Unicode Standard adopts the convention of placing the character U+200D ZERO WIDTH JOINER immediately after the encoded dead consonant. The ZERO WIDTH JOINER denotes a nonvisible letter that presents linking or cursive joining behavior on either side (that is, to the previous or following letter). Therefore, in the present context, the ZERO WIDTH JOINER may be considered to present a context to which a preceding dead consonant may join so as to create the half-form of the consonant. *[Unicode 4.0, p. 225.)]*

It should be noted here that not all consonants in Devanagari script have half forms. Unicode 4.0 is not explicit regarding the effect of ZWJ when used following such consonants, though it seems clear what the effect should be: a full conjunct-ligature should still be prevented, and given the absence of a half form, the only remaining alternative is to display the dead consonant with an overt halant.

As mentioned above, the formalisms specified in Section 9.1 are intended to be applied to all Indic scripts. Therefore, the use of ZWJ in a sequence < *consonant*, VIRAMA, ZWJ > to explicitly request a half form is considered to be applicable to all Indic scripts.

In terms of the relationship between the functions of VIRAMA and of ZWJ, the VIRAMA has the effect of killing the inherent vowel of the preceding consonant, creating a dead consonant. The effect of the ZWJ is to control how a dead consonant and a consonant following it will be presented.

3 Differences in conjoining strategies

At this point, an important observation must be made about the Devanagari non-ligature conjoining forms that have been described (the half forms and the reph): in a character sequence < C1, VIRAMA, C2 >, it is always the *dead* consonant, C1, that takes on the alternate form.

This is not always the case across Indic scripts, however. In many cases, it is the *live* consonant, C2, that takes on the alternate form. (I will hereafter refer to such behaviour as "C2-conjoining".) A C2-conjoining consonant typically takes an alternate form following a dead consonant, and is positioned below, below-right or to the right of the dead consonant. This is illustrated here using Oriya script:

Figure 1. Oriya below-base C2-conjoining form, "na-phalaa"

Figure 2. Oriya post-base C2-conjoining form, "yya-phalaa"

It is important to note the similarities and differences with the half forms and other C1-conjoining forms of Devanagari:

- For both C1- and C2 conjoining sequences, the dead consonant is C1.
- In C1-conjoining sequences, it is the dead consonant that takes the alternate form indicating the presences of a dead consonant. In contrast, in C2-conjoining sequences, the presence of a dead consonant is indicated by the *following* consonant taking an alternate form.

As mentioned above, Section 9.1 describes the encoding of Devanagari in detail, and presents that as a model that is followed for all other Indic scripts. There is a problem in this, however, in that not all Indic scripts use the same strategies for conjoining as does Devanagari. A summary of conjoining strategies (other than conjunct ligatures) for nine scripts is shown in Table 6.

Script	C1-conjoining forms		C2-conjoining forms	
	Reph	Half forms	Sub-base	Post-base
Devanagari	yes	most consonants	vattu (R)	
Bengali	yes	most consonants	B, R	Y
Gurmukhi			R, V, H	Y
Gujarati	yes	most consonants	vattu (R)	
Oriya	yes		most consonants	YY
Tamil				
Telugu			all consonants	
Kannada	yes		all consonants	
Malayalam	yes[4]	five "chillaksaram" [5]	L	Y, R, RR, V

Table 6. Summary of conjoining strategies by script

As can be seen here, Devanagari is not ideally suited to serve as a model for all scripts: it has half forms for most consonants but only one C2-conjoining consonant, and that one C2-

[4] Traditional Malayalam has a reph. It is not used in modern, reformed Malayalam, however.

[5] Malayalam has a set of five special forms, known as *chillaksaram*, in which the halant ligates with the dead consonant. These are like half forms in that C1 takes on a variant form, and it forms a cluster with the following consonant. The chillaksaram differ from half forms, however, in two respects. First, whereas half forms are generally the default when a conjunct ligature does not occur, chillaksaram are not used by default. Secondly, chillaksaram can be used in word-final position, whereas half forms cannot.

conjoining form, known as *vattu*, has special characteristics found elsewhere only in Gujarati.[6] In contrast, most Indic scripts do not have half forms. In fact, there are as many scripts that use half forms for most consonants (Devanagari, Gujarati, Bengali) as there are scripts that use C2-conjoining forms for most or all of their consonants (Oriya, Telugu, Kannada); and most scripts have multiple C2-conjoining forms.

The result, then, is that the general function of ZWJ in Indic scripts is defined in terms of a construct (the half form) that occurs in less than half the scripts. On the other hand, the effect of ZWJ in conjunct sequences used in most Indic scripts is not explicitly addressed. It is assumed that the principles described in Section 9.1 of Unicode 4.0 in relation to half forms can be applied to conjoining sequences of all Indic scripts in general, but this is problematic, as will be explained next.

4 Problems related to ZWJ in Indic scripts

4.1 ZWJ in C2-conjoining sequences

As described at the end of §2.2 of this document, in a sequence < C1, VIRAMA, ZWJ, C2 >, the virama indicates that C1 is dead, and the ZWJ affects how the dead consonant is rendered. Because the effects of both the virama and ZWJ co-locate in C1, without reference to C2, it is possible to use ZWJ to display a half form equally within a consonant cluster or in isolation:

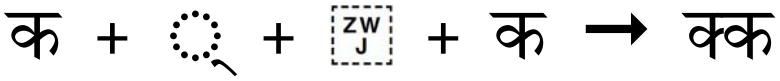

Figure 3. ZWJ used to display a half form Devanagari KA in consonant cluster"

Figure 4. ZWJ used to display a half form Devanagari KA in isolation"

The situation is quite different for C2-conjoining forms, however: for a script that uses C2-conjoining forms, a dead consonant never takes a different form in a non-ligated conjoining sequence. Therefore, it is meaningless to say that ZWJ can be used to display "a half form" in isolation:

[6] The special characteristic of the vattu is that it can occur as a sub-base form on the base consonant of a cluster, but also on a half form within a cluster of three or more consonants. Thus, in a Devanagari sequence < C1, virama, RA, virama, C2 >, the sub-sequence < C1, virama, RA > forms C1 with the sub-base vattu. This combination is then treated like a unitary consonant for purposes of conjoining with C2. For further information, see rule R8 in Section 9.1 of Unicode 4.0.

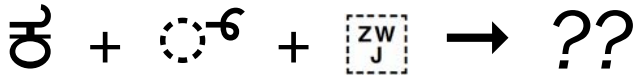

Figure 5. Undefined effect of ZWJ in an "isolated-half-form" sequence with Kannada KA"

For the same reasons, it is unclear what the effect of ZWJ should have in a consonant-cluster sequence for C2-conjoining cases:

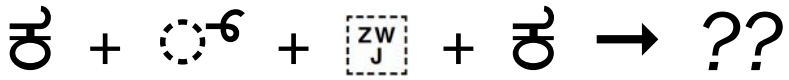

Figure 6. Undefined effect of ZWJ in an consonant-cluster sequence with Kannada KA"

Font and rendering-system implementers need to have a well-defined rendering for all potential sequences. In cases of ZWJ used in C2-conjoiing sequences, it is completely unclear what the effect of ZWJ on the rendering of the sequence should be.

4.2 Isolated C2-conjoining forms

From the preceding discussion, a significant gap is revealed related to the difference between C1- and C2-conjoining forms:-while there is an encoding mechanism for displaying half forms in isolation, no encoding mechanism has been specified for display of C2-conjoining forms in isolation.

The sequence < C, VIRAMA, ZWJ > is not appropriate to display the sub- or post-base conjoined form of a consonant C since this sequence represents a dead consonant, and for C2-conjoining consonants it is not the dead consonant that takes a sub- or post-base form. Moreover, this would create an inconsistency between that sequence and < C1, VIRAMA, ZWJ, C2 > since it would be completely inappropriate to display the latter as the sub-/post-base form of C1 followed by the full form of C2.

As users have requested a way to display sub- and post-base conjoining forms in isolation, an encoding mechanism for this is needed.

4.3 Explicit C2-conjoining forms rather than conjunct-ligature forms

Closely related to the display of conjoining forms in isolation is explicit encoding to request a non-ligated conjoining form rather than a conjunct ligature.

Conjoining of consonants in Indic scripts follows a three-level precedence hierarchy: a dead consonant C_d followed by a consonant C2 can be displayed in three ways:

1. the combination of C_d and C2 can form a conjunct ligature
2. either C_d or C2 takes on an alternate conjoining form and is combined with the full form of the other consonant
3. C_d is displayed with an overt halant, followed by the full form of C2

In general, the highest level available is used. Level 1 is used where a font supports a conjunct ligature for the given combination, but a font or rendering system can fallback to level 2 if a conjunct ligature is not supported or does not exist for the given combination. Likewise, fallback from level 2 to level 3 is possible when a conjoining form is not supported or does not exist.

Now, for all Indic scripts, ZWNJ can be used in a sequence < C1, virama, ZWNJ, C2 > to explicitly restrict the display to the level-3 alternative, the overt halant form.

For Indic scripts with half forms, ZWJ can be used in a sequence < C1, VIRAMA, ZWJ, C2 > to explicitly restrict the display to level 2 or level 3. There is a gap, however, in that no general mechanism is specified to provide comparable control in the case of C2-conjoining consonants.

It is assumed here that the solution for isolated C2-conjoining forms and for explicit request of conjoining forms rather than conjunct-ligature forms use identical sequences. That is, if a sequence χ is used to encode a C2-conjoining form in isolation, then the sequence < C1, χ > is used to explicitly request a C2-conjoining form rather than a conjunct-ligature form.

4.4 Ambiguous C1-/C2-conjoining sequences

In several Indic scripts, there are both consonants that have C1-conjoining forms, and consonants that have C2-conjoining forms. If a sequence < C1, VIRAMA, C2 > occurs in which C1 is a C1-conjoining consonant and C2 is a C2-conjoining consonant, then there may be ambiguity with regard to which consonant is displayed in an alternate form.

For Devanagari and Gujarati, there is only one consonant that has a C2-conjoining form: RA. In principle, there is a potential ambiguity in the case of sequences < C1, VIRAMA, RA > , but this is avoided since the rules in Section 9.1 make clear what the rendering should be:

- If C1 is RA, then C1 takes the reph form:

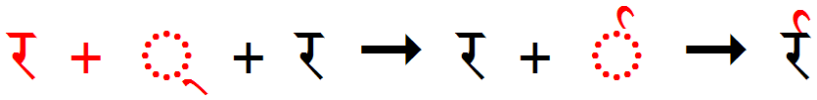

Figure 7. Specified rendering for Devanagari C1-/C2-conjoining sequence < RA, VIRAMA, RA >

- Otherwise, the C2-position RA takes the vattu form:

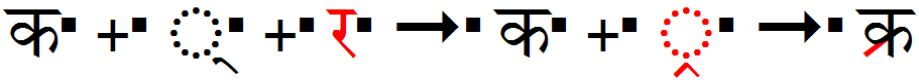

Figure 8. Specified rendering for other Devanagari C1-/C2-conjoining sequence

Beyond Devanagari and Gujarati, various scripts have both reph (a C1-conjoining form) and sub- or post-base C2-conjoining forms, and the rules in Section 9.1 can be applied as they are for Devanagari (shown above) to determine that, for a sequence < RA, VIRAMA, C2 >, the RA must display as reph rather than C2 taking on an alternate form. So far, then, the rules for Devanagari cover otherwise-ambiguous cases in other scripts.

There are exceptions to the general rules, however. Specifically, cases have been identified in which users need a sequence like < RA, VIRAMA, C2 > with the full form of RA and the sub- or post-base form

of C2, rather than the reph and full-form C2. In other words, there are two-way contrasts such as the following:

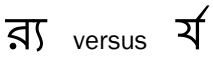

Figure 9. Bengali RA + ya-phalaa versus YA with reph

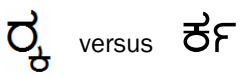

Figure 10. Kannada RA + ka-phalaa versus KA with reph

For distinctions such as these, a control character such as ZWJ could be used, but there is no general specification across Indic scripts for what encoded sequences to use to provide distinctions of this type. Instead, special-case rules are defined for particular cases in two scripts, Bengali and Kannada. The problem is that two different special-case mechanisms have been specified for two different scripts in order to resolve the same problem—one using ZWJ, and the other using ZWNJ:[7]

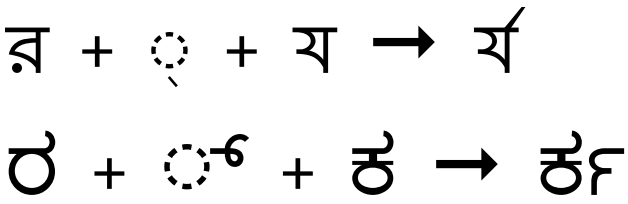

Figure 11. General Indic rule for reph formation applied to Bengali and Kannada clusters with C2-conjoining consonants

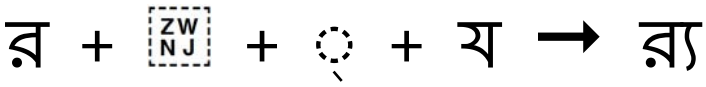

Figure 12. Special-case rule for Bengali RA + ya-phalaa

Figure 13. Special-case rule for Kannada RA + ka-phalaa

The problem, then, is that there is a category of distinction needing to be made in multiple scripts for which a general rule using a control character such as ZWJ could be used, but instead we have ad hoc rules that use different mechanisms.

Moreover, only two cases are covered, though there are potentially more cases of this type, such as Oriya < RA, VIRAMA, C2 >. Even if the other potential cases are not ones that users have identified as

[7] The text on p. 235 in Section 9.2 of Unicode 4.0 suggests different encoded representations for Bengali than are shown in Figure 11 and Figure 12. That specification in Unicode 4.0 contradicted the specification indicated by the rules in Section 9.1, however, and was subsequently revised in Unicode 4.0.1. See http://www.unicode.org/versions/Unicode4.0.1/ for details.

needing a two-way contrast, rendering implementations must produce *some* output for any potential sequence, and general rules in such cases applied consistently across scripts are much better for implementers and for users than a patchwork of ad hoc, special cases.

5 What is needed

In light of the preceding discussion, it would be beneficial to have the following points addressed:

- Define the intended effect of ZWJ on rendering of the sequences < C1, VIRAMA, ZWJ > and < C1, VIRAMA, ZWJ, C2 > for C2-conjoining cases.
- Specify an encoding mechanism that can be used to display sub- or post-base C2-conjoniing forms in isolation, or to explicitly request that a C2-conjoining form be used in lieu of a conjunct-ligature form.
- Establish general rules applied consistently across all scripts that permit explicit control for two renderings in the otherwise-ambiguous cases of < C1, VIRAMA, C2 > in which C1 is a C1-conjoining consonant and C2 is a C2-conjoining consonant.

6 Possible solutions

The different problems that have been identified are interrelated, and so a solution adopted for one will affect the solutions adopted for others. If we consider possible solutions for explicitly requesting a C2-conjoining form, each possibility will largely determine how the other problems must be handled. The possible solutions to that problem, therefore, will be taken as a starting point.

I will assume that the encoded sequence used for explicitly requesting a C2-conjoining form rather than a conjunct-ligature form (i.e., limiting the possible rendering of a consonant cluster to levels 2 or 3 as described in §4.3) must use a character sequence that involves VIRAMA and ZWJ, just as for C1-conjoining consonants, rather than some other sequence. This leaves two possibilitites:

< C1, VIRAMA, ZWJ, C2 >

or

< C1, ZWJ, VIRAMA, C2 >

For each possibility, a rationale can be suggested. Each will be considered in turn.

6.1 The < VIRAMA, ZWJ > solution

The representation for explicit C2-conjoining forms using VIRAMA + ZWJ maintains the same linguistic or functional significance of characters as in the case of C1-conjoining consonants: the virama in the sub-sequence < C1, VIRAMA > indicates that the inherent vowel of C1 is killed, resulting in a dead consonant; and the ZWJ affects how the dead consonant and the following consonant conjoin, as shown in Table 7:

Sequence	Possible renderings (in preferential order)
C_d + C	• conjunct ligature • sub-/post-base C2-conjoining form • C_d with overt halant + C
C_d + ZWJ + C	• sub-/post-base C2-conjoining form • C_d with overt halant + C

Table 7. Effect of ZWJ between C_d and C limiting possible renderings

Maintaining the principle that the sequences used for explicit request of a C2-conjoining form within a consonant cluster and in isolation are the same, we arrive at the following renderings for different sequences:

Sequence	Rendering
< C1, VIRAMA, ZWJ, C2 >	C1 + sub-/post-base form of C2
< C1, VIRAMA, ZWJ >	C1 with overt halant
< SPACE, VIRAMA, ZWJ, C2 >[8]	isolated sub-/post-base form of C2

Table 8. Implication of < VIRAMA, ZWJ > solution

While the < VIRAMA, ZWJ > solution is tidy with respect to the meaning and function of the VIRAMA and ZWJ characters in different contexts, it has some shortcomings.

First of all, it leaves unresolved the problem of ambiguous C1-/C2-conjoining sequences described in §4.4. For instance, consider a sequence < C1, VIRAMA, ZWJ, C2 >: it is unclear whether the effect of the ZWJ should be to explicitly request a half form of C1 or a sub- or post-base form of C2. Note also that in one potential case requiring disambiguating contrast, the Devanagari RA, the sequence < RA, VIRAMA, ZWJ, RA > has been specified in Section 9.1 to display the "eyelash" C1-conjoining form of ra. At best, then, there will be inconsistencies: < C1, VIRAMA, ZWJ, C2 > will produce a C1-conjoining form in some but not all cases for a given value of C1, and a C2-conjoining form in some but not all cases for a given value of C2.

Secondly, the sub-sequence < VIRAMA, ZWJ, C2 > does not lend itself to display modes in which control-picture glyphs for non-printing characters are displayed. The reason for this is that, operationally, a rendering system must match the combination of virama and C2 as substitute the conjoining form of C2. Consider an example using Oriya, and suppose the control-picture glyph for ZWJ is ◌‍. then the display of different kinds of sequences with or without showing non-printing control characters would be as illustrated in Table 9:

[8] A space is added to the sequence for an isolated conjoining form since the sub-base form is a mark, and a mark requires some character to act as a base.

Sequence	Normal display	Possible display with control-picture glyphs shown
< KA, VIRAMA, ZWJ, RA >		
< SPACE, VIRAMA, ZWJ, RA >		

Table 9. Rendering using control-picture glyph for ZWJ

The sharp contrast in rendering between display modes would likely be confusing to users. Users would likely be confused or perhaps even misled by the rendering of the display-controls mode due to the unclear relationship between what is displayed and the underlying character sequences.

Such issues do not arise for sequences < C, VIRAMA, ZWJ, … > in which C is a C1-conjoining consonant since the ZWJ does not occur between the consonant and virama that operationally merge to form the conjoining form. As will be seen below, such problems do not arise for C2-conjoining cases using the alternate solution.

6.2 The < ZWJ, VIRAMA > solution

The alternate solution, representing explicit C2-conjoining forms using ZWJ + VIRAMA, is not as tidy in terms of abstract conceptualizations of the linguistic meaning and function of the ZWJ and VIRAMA characters since, in a sequence < C1, ZWJ, VIRAMA, C2 >, the virama kills the vowel of the preceding consonant, but the ZWJ comes between the virama and consonant. Conceptually, we would have to imagine that < C1, ZWJ, VIRAMA > results in an alternate form of dead C1 that has different conjoining behaviour.

Operationally, however, this solution has advantages. In a sequence < C1, VIRAMA, C2 > in which C2 is a conjoining consonant, although the virama is logically associated with C1, in terms of rendering operations, the virama must be associated with C2. That is, a rendering operation will detect the combination of VIRAMA + C2 and will substitute the conjoining form of C2. Because of this, inserting ZWJ before VIRAMA is better suited to processing since it does not interfere in the VIRAMA + C2 sequence.

The effect of ZWJ on conjoining would be the same as in the alternate solution; it is only the sequences involved that differ:

Sequence	Possible renderings (in preferential order)
< C1, VIRAMA, C2 >	• conjunct ligature • sub-/post-base C2-conjoining form • C1 with overt halant + C2
< C1, ZWJ, VIRAMA, C2 >	• sub-/post-base C2-conjoining form • C1 with overt halant + C2

Table 10. Effect of ZWJ in < ZWJ, VIRAMA > solution

Implications of this solution for rendering of various other sequences are as follows:

Sequence	Rendering
< C1, VIRAMA, ZWJ, C2 >	C1 with overt halant + C2
< C1, VIRAMA, ZWJ >	C1 with overt halant
< C1, ZWJ, VIRAMA, C2 >	C1 + sub-/post-base form of C2
< SPACE, ZWJ, VIRAMA, C2 >	isolated sub-/post-base form of C2

Table 11. Rendering of various sequences in < ZWJ, VIRAMA > solution

This solution also provides a solution for the problem of ambiguous C1-/C2-conjoining sequences described in §4.4. Whether for the cases for which there is a known need, in which C1 is RA, or more generally, the sequence specified here for C2-conjoining forms and the sequence specified for half forms in Section 9.1 of Unicode 4.0 provide for any distinction that might be needed. The relevant sequences and their corresponding rendering is shown in Table 12

Sequence	Rendering
< RA, VIRAMA, C2 >	reph on C2
< RA, VIRAMA, ZWJ, C2 >	half form of RA ("eyelash RA") + C2
< RA, ZWJ, VIRAMA, C2 >	full RA + sub-/post-base form of C2
< C1, VIRAMA, C2 >	default rendering according to script, combination of consonants and font (conjunct ligature, half C1 + full C2, or full C1 + sub-/post-base C2)
< C1, VIRAMA, ZWJ, C2 >	half form of C1 + full C2
< C1, ZWJ, VIRAMA, C2 >	full C1 + sub-/post-base form of C2

Table 12. Disambiguation of sequences in < ZWJ, VIRAMA > solution

The < ZWJ, VIRAMA > solution also avoids problems with display modes that show control-picture glyphs for control characters. Consider again an example using Oriya:

Sequence	Normal display	Possible display with control-picture glyphs shown
< KA, ZWJ, VIRAMA, RA >		
< SPACE, ZWJ, VIRAMA, RA >		

Table 13. Rendering using control-picture glyph for ZWJ

In this case, the relationship between what is displayed and the underlying character sequences is predictable, and the relationship between what is seen in the two display modes is clear. Therefore, this would likely be less confusing for users than the alternate solution.

7 Proposal

In view of the relative merits of the two alternate solutions for encoding of explicit C2-conjoining forms, it is proposed that the second alternative, the < ZWJ, VIRAMA > solution, be adopted. More generally, the following is proposed for a complete specification of the functions of ZWJ and ZWNJ in relation to consonant sequences in Indic scripts.

Conjoining of consonants in Indic scripts follows a three-level precedence hierarchy: a dead consonant C_d followed by a consonant C2 can be displayed in three ways:

1. the combination of C_d and C2 can form a conjunct ligature
2. either C_d or C2 takes on an alternate conjoining form and is combined with the full form of the other consonant
3. C_d is displayed with an overt halant, followed by the full form of C2

In general, the highest level available is used. Level 1 is used where a font supports a conjunct ligature for the given combination, but a font or rendering system can fallback to level 2 if a conjunct ligature is not supported or does not exist for the given combination. Likewise, fallback from level 2 to level 3 is possible when a conjoining form is not supported or does not exist.

For a sequence < RA, VIRAMA, C >, if the script in question has a reph form, this takes precedence over other conjoining forms. Otherwise, the default rendering for a sequence < C1, VIRAMA, C2 > depends on a combination of the script, the particular consonants involved, and the font.

The characters ZWJ and ZWNJ can constrain the possible renderings as follows:

- For all Indic scripts, ZWNJ can be used in a sequence < C1, virama, ZWNJ, C2 > to explicitly restrict the display to the level-3 alternative, the overt halant form. No other function for ZWNJ is defined.
- For C1 a C1-conjoining consonant, ZWJ can be used in a sequence < C1, VIRAMA, ZWJ, C2 > to restrict the display to level 2 or level 3. Specifically, this sequence requests the half form of C1, to be combined with the full form of C2. If C1 has no half form, then fallback to the level 3 display is used.
- For C2 a C2-conjoining consonant, ZWJ can be used in a sequence < C1, ZWJ, VIRAMA, C2 > to restrict the display to level 2 or level 3. Specifically, this sequence requests the sub- or post-base form of C2, to be combined with the full form of C1. If C2 has no sub- or post-base form, then fallback to the level 3 display is used.
- For a C1-conjoining consonant, the sequence < C, VIRAMA, ZWJ > can be used to display the half form of C in isolation.
- For a C2-conjoining consonant, the sequence < SPACE, ZWJ, VIRAMA, C > can be used to display the sub- or post-base form of C in isolation.[9]

This specification covers all of the scenarios outlined in §5. It is proposed that this specification would be applied consistently across all South-Asian scripts; that is, Devanagari, Bengali, Gurmukhi, Gujarati, Oriya, Tamil, Telugu, Kannada, and Malayalam scripts.[10]

---

[9] It might also work to specify a sequence < SPACE, VIRAMA, C > for display of a sub- or post-base form in isolation. Whereas with explicit half forms, the ZWJ is needed to provide contrast with overt-halant forms, a two-way distinction is not needed for C2-conjoining forms; thus, there is no specific reason to use ZWJ other than to maintain symmetry with the sequence used to display half forms in isolation.

In order to make this applied consistently across all ten scripts, it would be necessary to revise current specifications for Kannada, in relation to RA + conjoining C, and Bengali, in relation to RA + ya-phalaa:

Kannada text element	Currently specified sequence	Revised specification
ಕ್ರ	< RA, VIRAMA, KA >	(same)
ರ್ಕ	< RA, VIRAMA, ZWJ, KA >	< RA, ZWJ, VIRAMA, KA >

Table 14. Revised sequence for Kannada RA + conjoining C

Bengali text element	Currently specified sequence	Revised specification
র্য	< RA, VIRAMA, YA >	(same)
র‍্য	< RA, ZWNJ, VIRAMA, YA >	< RA, ZWJ, VIRAMA, YA >

Table 15. Revised sequence for Bengali RA + ya-phalaa

[10] Some variation on this specification would be required for Sinhala script, due to the fact that the default rendering for sequences < C, VIRAMA, C > is *not* to display a conjunct ligature (if it exists) but rather to display an overt-halant (al-lakuna) form. It would be beneficial if the function of ZWJ and ZWNJ can be kept as consistent as possible for Sinhala as for other Indic scripts, though it must be recognized that they cannot be entirely the same.