

Michael Kaplan
November 7, 2000

Problems with the Tamil Block Description

Tamil has been included in Unicode since 2.0, and no actual changes have been made to it between 2.x and 3.0x. There is, however, a specific problem in regards to the block description. It can be found in the discussion of *Ligatures* (p.230-232), in particular #5 on page 232, which reads:

5. The vowel sign AI changes to <elpehant trunk> to the left of NNA, NNNA, LA, or LLA.

It then gives several examples. The elephant trunk is a description of the glyph seen below:



Essentially, this glyph is intended to replace the vowel sign AI, which should be reordered to appear visually before any character it logically follows.

The problem is that this particular rendering, often referred to as ORNL (which stands for the Old style R, N, and L) is not used at all in modern Tamil, whether in Tamilnadu, Sri Lanka, Singapore, Malaysia, or in large other populations such as those in Canada. It is also not used in other languages that make use of the Tamil script like Bagada.

Companies and individuals that have implemented and released fonts that support Tamil (including Microsoft, Agfa Monotype, and James Kass) have pretty much ignored this point in the block description and instead just done the typical reordering that one expects with AI. Since this matches modern usage, it certainly is a good thing that they all do this. However, it would be good for Unicode to match what users, and its own members, are currently implementing. It is also of note that neither TSCII nor TAB (the two "sort of" competing standards for 8-bit Tamil encodings) support the ORNL rendering.

Only Unicode and ISCII describe it (and some random fonts include the glyph, not mapped to any standard).

My (extreme) proposal is to remove #5 entirely. This would match current usage and the way that every font I could find is currently implementing the Tamil script.

On the assumption that people within the UTC might consider this a bit too drastic, and also in recognition of the fact that there are traditional documents that would expect the ORNL rendering, my less drastic proposal is to make use of either the ZERO WIDTH JOINER (U+200D) or possibly the Grapheme Joiner (also on the agenda) to support the less-used, traditional form. Thus, the rendering would be as follows:

Logical		Rendered
U+0BA3 U+0BC8	NNA + AI	ಣೆಣ
U+0BA9 U+0BC8	NNNA + AI	ಣೆ
U+0BB2 U+0BC8	LA + AI	ಲೆ
U+0BB3 U+0BC8	LLA + AI	ಲೆ
U+0BA3 U+200D U+0BC8	NNA + ZWJ + AI	ಣೆ
U+0BA9 U+200D U+0BC8	NNNA + ZWJ + AI	ಣೆ
U+0BB2 U+200D U+0BC8	LA + ZWJ + AI	ಲೆ
U+0BB3 U+200D U+0BC8	LLA + ZWJ + AI	ಲೆ

The ZWJ solution was suggested by several people on the Unicode list, however none of the fonts that currently exist seem to implement ZWJ as these people implied they should, which would be to continue to support the reordering. In all of the fonts I would try, inserting U+200D (ZWJ) or U+200B (zero-width space or ZWSP) would stop the reordering. Rick McGowan, Marco Cimarosti, and Antoine Leca all believe that this *should* work, which suggests that this would be incorrect behavior on the part of the fonts.

Whether using the grapheme joiner here is something that could be discussed. The purpose would be to alter the joining behavior without affecting other rules (such as re-ordering).

Summary

One of two proposed solutions:

- 1) Remove #5 on page 317 from TUS, or
- 2) Replace the current #5 with a description including use of the ZWJ or grapheme joiner.