Fixing Indic2 OpenType Layout

John Hudson, Tiro Typeworks Ltd., 28 April 2014, version 1.1

This recommendation paper follows from earlier documents Problems for Indic typography in current OpenType Layout implementations *and* Enabling Typography: towards a general model of OpenType Layout. *This paper represents the consensus reached in meetings of* OTL *engine developers in Seattle, 21–25 April 2014.*

The term 'Indic2' refers to existing implementations of the Microsoft's Indic script shaping model using the following tags: <bng2> <dev2> <gjr2> <gur2> <knd2> <mlm2> <ory2> <tml2> *and* <tel2>. *That is, the model introduced in Windows Vista and implemented in Uniscribe and compatible shaping engines. The recommendations in this paper could also be applied to fix the same issues in Indic1* (<beng> <deva> etc.) *layout engines that use the old Windows* XP *shaping model, but since they require updates to the engines, best results would involve switching to an Indic2 model.*

Among topics of discussion at the meetings in Seattle was an Indic3 shaping model making use of a more powerful and flexible generic engine derived from proposed Unicode Indic properties. More information about this will be provided in future documents; this paper is focused on fixing existing Indic layout implementations.

Background

As discussed and illustrated in the earlier document on *Problems for Indic typography in current OpenType Layout implementations,* Uniscribe-compatible layout engines apply all GSUB features listed in the Microsoft Indic font specifications to discrete orthographic syllables ('clusters'). This means that not only the 'basic shaping forms' features that are required to be applied at the cluster level and which affect reordering, but also the 'mandatory presentation form' and contextual alternate features are applied to segmented clusters.[1] This means that these presentation form and contextual features cannot be used to perform substitutions that rely on cross-cluster glyph sequences or contexts, effectively disabling typographic mechanisms at the font level to resolve relationships of adjacency at cluster boundaries. This can result in collisions between adjacent shapes, crowding and other disruptions of readable text.

The obvious fix for this situation would be to change existing shaping engines so that the mandatory presentation forms and all other post-reordering features apply to whole glyph runs rather than to discrete clusters. That would enable existing presentation form layout features such as

1. In the general model for OpenType Layout proposed recently, the 'basic shaping form' features of Microsoft's Indic specifications are classified as 'orthographic unit shaping'; the 'mandatory presentation form' and all other post-reordering GSUB features are classified as 'typographic layout'. Orthographic unit shaping features apply, in Indic scripts, to discrete clusters; typographic layout features *should* apply to complete glyph runs, ignoring cluster boundaries.

<pres> <abvs> <blws> etc.—which are in any case applied simultaneously by the layout engine and rely on font lookup ordering to determine their effect—to include cross-cluster input sequences and context statements. However, testing of this approach in the Harfbuzz layout engine indicates that this change would break display of some fonts, which include lookups that are intended to apply only within clusters but that would trigger substitutions across cluster boundaries if applied to whole glyph runs. It is unfortunate that fonts have been made in this way, but given the behaviour established by Uniscribe it is difficult to consider them malformed per se. This constitutes a backwards compatibility requirement. Therefore, the recommendations outlined below seek to provide a path to fixing Indic2 shaping in layout engines and fonts using these principles:

1. The fix should be reasonably easy to apply, both in shaping engines and in fonts.
2. Deployed together, updated layout engines and fonts should provide a staightforward mechanism for affecting cross-cluster substitutions and contexts.
3. Within reasonable expectations, updated fonts used with non-updated layout engines should fail in the same way as non-updated fonts (*i.e.* cross-cluster substitutions and contexts will not be applied).
4. Within reasonable expectations, non-updated fonts used with updated layout engines should fail in the same way as with non-updated engines (*i.e.* cross-cluster substitutions and contexts will not be applied).

The caveat 'within reasonable expectations' is necessary because it would be relatively easy to build a font with GSUB lookups that would break the backwards compatibility provisions of the following recommendations. The provisions target the typical font implementations that have been observed in testing.

Recommendations

In order to provide for backwards compatibility, layout features currently applied to discrete clusters must continue to be applied in this way. This means that GSUB lookups to affect cross-cluster substitutions and contexts must be associated with different layout features, which must then be applied to whole glyph runs rather than discrete clusters. This means that the following features—a superset of those referenced in the Microsoft Indic font specifications—should be applied to discrete clusters:

<nukt>	Nukta Forms
<akhn>	Akhands
<rphf>	Reph Forms
<pref>	Pre-base Forms
<blwf>	Below-base Forms
<half>	Half Forms
<vatu>	Vattu Forms
<rkrf>	Rakar Forms
<cjct>	Conjunct Forms
<pstf>	Post-Base Forms

Reordering occurs here

<pres>	Pre-base Substitutions
<abvs>	Above-base Substitutions
<blws>	Below-base Substitutions
<psts>	Post-base Substitutions
<haln>	Halant Forms
<calt>	Contextual Alternates

Ideally, in order to provide the most flexibility to Indic typography, all other layout features—both GSUB and GPOS—, should be applied to whole glyph runs, ignoring cluster segmentation. However, to provide a reliable and predictable mechanism for cross-cluster substitutions and contexts, such lookups should to be associated with the layout feature

<rclt>	Required Contextual Alternates

and this feature must be applied to whole glyph runs by the fixed layout engine. Note that there should be no restriction assumed on the types of lookups associated with this feature.

It may be noted by font developers that, if targeting only at layout engines fixed according to these recommendations, it would be possible to associate all presentation form substitution lookups in a font with the <rclt> feature. However, to be compatible with older versions of layout engines, which might be difficult to update in some environments, it is recommended that *only* lookups involving cross-cluster substitutions and contexts should be associated with this feature. All presentation form substitutions affecting discrete clusters should continue to be associated with <pres> <abvs> <blws> etc.. This means that font developers will need to carefully plan behaviours and design lookup structures accordingly.

A note on the <init> topographical feature

Uniquely, the Microsoft Bengali specification calls for the

<init>	Initial Forms

feature to be applied during basic shaping form substitutions. This feature typical substitutes the non-left-connecting form of the independent vowels ে and ৈ. This feature relies on character string analysis and, as specified, should apply only to the first character in a word. In fixed Indic2 layout engines and fonts, this feature should continue to be applied as it has been to date.

Important

Although the recommendations in this document represent the consensus of shaping engine developers currently employed by Adobe, Apple, Google, and Microsoft, this document does not constitute a formal commitment to implementing these recommendations in any specific product or products. Those attending the meetings in Seattle did so as invited expert individuals, and not as formal representatives of their respective employers.