

Proposal to redefine OpenType Layout joining features

Draft 1.2, 3 February 2016

John Hudson, Tiro Typeworks Ltd

Rationale

The four OTL glyph substitution features `<isol>`, `<init>`, `<medi>` and `<fina>`—variously referred to as ‘topographical’, ‘positional’, or ‘joining’ form features—were registered in the early days of OpenType. The latter three were registered jointly by Microsoft and Adobe, and `<isol>` by Microsoft alone. Features are not date stamped, but I suspect that the `<isol>` feature was added after the others, when developers at Microsoft realised that there were some circumstances in which the isolated presentation form of a character might differ from its default, cmap-encoded form.

The features as originally registered and defined are typified by the `<init>` feature description:

Friendly name: Initial Forms

Registered by: Microsoft/Adobe

Function: Replaces glyphs at the beginnings of words with alternate forms designed for this use. This is common in Latin connecting scripts, and required in various non-Latins like Arabic.

Example: In the typeface Ex Ponto, the default t in the word ‘type’ is replaced with the t.begin form.

Recommended implementation: The init table maps default alphabetic forms to corresponding beginning forms (GSUB lookup type 1).

Application interface: The application is responsible for noting word boundaries. For GIDs at the beginnings of words and found in the init coverage table, the application passes a GID to the feature and gets back a new GID.

UI suggestion: This feature should be active by default.

Script/language sensitivity: Can be used in any alphabetic script.

Feature interaction: This feature may be used in combination with other substitution (GSUB) features, whose results it may override. See also `<medi>` and `<fina>`.

As with many of the early OTL feature descriptions, there is much in this that is speculative: it represents how some people at a particular point in time imagined that a technology might work. In the event, these four features turned out to work almost exactly *not* as described, either in layout engines or in fonts. For this reason, it was agreed at the recent OT ad hoc working group meeting (OTWG2, 29–30 October 2015, Mountain View) that the actual implementation of these features should be clarified and restricted, both to better represent how they currently work and to provide an opening for potential new features to address some aspects of the original speculation not handled by these features in practice. [This proposal does not include such new features, which require additional discussion. The purpose of the present proposal is to clarify the existing features.]

The example given in the feature description is for a Latin cursive style typeface, Ex Ponto (an Adobe Original, designed by Jovica Veljović), whose forms are imagined being replaced based on their position in a word, as determined by the application ‘responsible for noting word boundaries’. In fact, this doesn’t happen. At the OTWG meeting, Miguel Sousa (Adobe) confirmed that neither Ex

Ponto nor other Adobe cursive Latin fonts are built in this way, and Eric Muller (Amazon, formerly Adobe) pointed out that the Bengali <init> legacy implementation (see below) is in fact the only place in OpenType Layout where engines are expected to identify word boundaries for glyph processing purposes. I am aware of only one implementation that seeks to apply the features as currently described: a little-known, non-default option in Adobe InDesign.

The four features are, however, applied in shaping of Arabic and similar scripts with standard joining behaviour specified in the Unicode Standard; however, they are not applied as described in the registered features. Arabic shaping does not rely on word boundary identification at the application level, but on the joining property of individual characters specified in the Unicode [<ArabicShaping.txt>](#) standard. [Note that this document, despite its name, specifies joining behaviours for characters in all joined scripts in Unicode, not just Arabic.] This is essential for such scripts, because form selection is not based on word position but on the properties of adjacent characters. Isolated, initial and final forms of Arabic letters, for example, may all occur in the middle of words, *e.g.* الإمارات (*al-imārāt*).

There is one documented instance in which one of the joining features, <init>, is required to be activated at the beginning of a word, and that is in the Bengali shaping models for the [<beng>/<bng2>](#) script tags. As noted by Eric Muller, this is currently the only instance in which there is a clear expectation of an OTL engine needing to determine a word boundary in text for the purposes of orthographic shaping. For the purposes of this proposal, this singular case is to be recorded as a legacy anomaly. A potential <bng3> script tag for Universal Shaping Engine processing would not specify this feature, and instead Bengali initial forms would be handled contextually in GSUB lookups, as other positional variants in that script must be already, and as are handled in other scripts. [Alternatively, true word-positional features could be registered to behave as initially imagined the joining features would, independently of Unicode joining properties, relying on layout engine identification of word boundaries, with caveats about the non-triviality or inappropriateness of that for some writing systems. If desirable, mechanisms for addressing other aspects of text topography could be considered, such as beginning/end-of-line forms, top/bottom-of-text-block, etc.. Being independent of the ArabicShaping.txt-related features, such mechanisms would also be available to Arabic and other such scripts as rely on the existing features for orthography shaping. Again, the current proposal does not include anything along these lines.]

The registered OpenType Layout feature set also includes an additional shaping feature <med2> ‘Medial Forms #2’ whose use is restricted to the Syriac script. Arguably this feature is unnecessary, as the substitution involved could be handled contextually at the glyph level, *e.g.* with the <rclt> Required Contextual Forms feature, but the present proposal does not suggest any change to the feature description or implementation of this feature.

Proposal

For these reasons, it is proposed to revise the registered descriptions of the joining features to

- a) remove all reference to word boundary identification and associated application of features;
- b) explicitly redefine these features in terms of implementation of Unicode [<ArabicShaping.txt>](#);
- c) remove all examples not involving Unicode [<ArabicShaping.txt>](#) and replace with appropriate examples.

The proposed revised wording of the registered feature descriptions follow.

Tag: 'fina'

Friendly name: Terminal Forms

Registered by: Microsoft/Adobe (revised 2016, OTWG2)

Function: Implements Unicode [<ArabicShaping.txt>](#) orthographic shaping for single-joining final ('R Right_Joining' in right-to-left scripts; 'L Left_Joining' in left-to-right scripts) or dual-joining characters in final situation.

Example: In an Arabic script font, the default form of the letter **g** is replaced with the right-joining final form **g** when following a left-joining character.

Recommended implementation: The 'fina' table maps default forms to corresponding single-joining final forms (GSUB lookup type 1). Contextual lookups (GSUB lookup type 5, 6 or 8) may also occur in some fonts.

Application interface: The application is responsible for parsing character strings and identifying which orthographic shaping form feature to apply to which GIDs, based on the character `Joining_Type` property specified in the Unicode [<ArabicShaping.txt>](#) standard (additional factors, such as the presence of control characters, may determine a final situation). For GIDs with single-joining final forms found in the 'fina' coverage table, the application passes a GID to the feature and gets back a new GID.

UI suggestion: This feature should be active by default.

Script/language sensitivity: Can be used in any script with joining properties specified in [<ArabicShaping.txt>](#).

Feature interaction: This feature may be used in combination with other substitution (GSUB) features, whose results it may override. See also 'init', 'isol', and 'medi'.

Tag: 'init'

Friendly name: Initial Forms

Registered by: Microsoft/Adobe (revised 2016, OTWG2)

Function: Implements Unicode [<ArabicShaping.txt>](#) orthographic shaping for single-joining initial ('R Right_Joining' in left-to-right scripts; 'L Left_Joining' in right-to-left scripts) or dual-joining characters in initial situations.

Example: In an Arabic script font, the default form of the letter **س** is replaced with the left-joining initial form **س** when preceding a right-joining character.

Recommended implementation: The 'init' table maps default forms to corresponding single-joining initial forms (GSUB lookup type 1). Contextual lookups (GSUB lookup type 5, 6 or 8) may also occur in some fonts.

Application interface: The application is responsible for parsing character strings and identifying which orthographic shaping form feature to apply to which GIDs, based on the character `Joining_Type` property specified in the Unicode [<ArabicShaping.txt>](#) standard (additional factors, such as the presence of control characters, may determine an initial situation). For GIDs with single-joining initial forms found in the 'init' coverage table, the application passes a GID to the feature and gets back a new GID.

UI suggestion: This feature should be active by default.

Script/language sensitivity: Can be used in any script with joining properties specified in [<ArabicShaping.txt>](#).

Feature interaction: This feature may be used in combination with other substitution (GSUB) features, whose results it may override. See also 'fina', 'isol', and 'medi'.

Tag: 'isol'

Friendly name: Isolated Forms

Registered by: Microsoft/Adobe (revised 2016, OTWG2)

Function: Implements Unicode [<ArabicShaping.txt>](#) orthographic shaping for non-joining or left-, right-, or dual-joining characters in isolated situation. Note that in many fonts this feature may not be implemented, as the default forms of the relevant characters are the isolated forms; in some fonts this feature may involve contextual substitution based on kind of isolated situation.

Example: In an Arabic script font, the default form of the letter ا is replaced with a special isolated form ا when no adjacent to any other letter or numeral (this would be a contextual substitution).

Recommended implementation: The 'isol' table maps default forms to corresponding isolated forms (GSUB lookup type 1). Contextual lookups (GSUB lookup type 5, 6 or 8) may also occur in some fonts.

Application interface: The application is responsible for parsing character strings and identifying which orthographic shaping form feature to apply to which GIDs, based on the character Joining_Type property specified in the Unicode [<ArabicShaping.txt>](#) standard (additional factors, such as the presence of control characters, may determine an isolated situation). For GIDs with isolated forms found in the 'isol' coverage table, the application passes a GID to the feature and gets back a new GID.

UI suggestion: This feature should be active by default.

Script/language sensitivity: Can be used in any script with joining properties specified in [<ArabicShaping.txt>](#).

Feature interaction: This feature may be used in combination with other substitution (GSUB) features, whose results it may override. See also 'fina', 'init', and 'medi'.

Tag: 'medi'

Friendly name: Medial Forms

Registered by: Microsoft/Adobe (revised 2016, OTWG2)

Function: Implements Unicode [<ArabicShaping.txt>](#) orthographic shaping for dual-joining characters in medial situations.

Example: In an Arabic script font, the default form of the letter ق is replaced with the left-joining initial form ق when and following a left-joining character and preceding a right-joining character.

Recommended implementation: The 'init' table maps default forms to corresponding dual-joining forms (GSUB lookup type 1). Contextual lookups (GSUB lookup type 5, 6 or 8) may also occur in some fonts.

Application interface: The application is responsible for parsing character strings and identifying which orthographic shaping form feature to apply to which GIDs, based on the character Joining_Type property specified in the Unicode [<ArabicShaping.txt>](#) standard (additional factors, such as the presence of control characters, may determine a medial situation). For GIDs with dual-joining forms found in the 'medi' coverage table, the application passes a GID to the feature and gets back a new GID.

UI suggestion: This feature should be active by default.

Script/language sensitivity: Can be used in any script with joining properties specified in [<ArabicShaping.txt>](#).

Feature interaction: This feature may be used in combination with other substitution (GSUB) features, whose results it may override. See also 'fina', 'init', and 'isol'.