L2/06-332

Title: Proposal to change the Bidi category of five Arabic characters from AL to AN

Requesters: Behdad Esfahbod, GNOME Foundation
Roozbeh Pournader, Sharif FarsiWeb, Inc.

Date: 2006-10-15

This proposal is to ask for changing the previous bidirectional categories of five Arabic characters. This proposal is made in order to help implementers of the Unicode standard use or re-use existing code and mechanisms available for rendering the Arabic script as much as possible instead of forcing them to handle these marks quite differently. We believe that with "smart-font" technologies like OpenType, AAT, and SIL Graphite, these characters will not really need special handling from the rendering engine, if the proposed change is accepted.

The problems one of the authors of this proposal has encountered in implementing support for the five mentioned characters in Pango[1] has led us to believe that changing the Bidi categories of these five characters to a category that is more compatible with numbers makes them more easily implementable.

The characters discussed are U+0600 ARABIC NUMBER SIGN, U+0601 ARABIC SIGN SANAH, U+0602 ARABIC FOOTNOTE MARKER, U+0603 ARABIC SIGN SAFHA, and U+06DD ARABIC END OF AYAH. All these five characters precede a sequence of digits that appear after them, and interact with them. All the characters are presently characters of General Category "Cf" (Other, Format) and Bidi Category "AL" (Right-to-Left Arabic). Four of them (U+0600..U+0603) extend under the sequence of digits, and the other (U+06DD) encloses them. For more details about the behavior of the characters, see *The Unicode Standard 4.0*, Section 8.2, pp 198–199.

Figure 1 shows the details of the processing of the character sequence ⟨U+0600, U+0661, U+0661, U+0662⟩ by a Unicode text rendering engine. First, the characters are assigned bidirectional levels according to the UAX #9 The Bidirectional Algorithm, then their directional runs are determined and the characters are ordered according to the runs, and finally, the sequence is magically rendered according to the specified Unicode behavior.

1 Pango is a Unicode and OpenType text rendering engine commonly used in GNU/Linux; see http://www.pango.org/ for more information

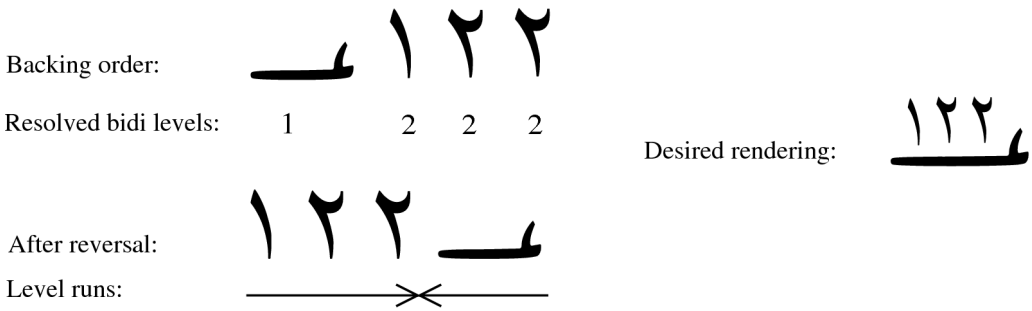

Figure 1: Processing details of a common character sequence that includes U+0600 ARABIC NUMBER SIGN and a digit sequence

Most of the existing Unicode text rendering engines, including Pango, simply break text into bidirectional runs, then process them separately, and finally put them back together or show them next to each other. This is based on the recommendation of the UAX #9, The Bidirectional Algorithm, Section 3.5 that states that shaping "should remain limited to characters within the same directional run".

This means that in rendering the character sequences that include the Arabic characters under discussion (like in the ARABIC NUMBER SIGN example in Figure 1), the rendering engines render the special character and the sequence of digits after it separately. So, in the case the font contains special glyph substitution information for the special character(s) involved (like the SIL font Scheherazade that implements such features as OpenType GSUB tables), they will be simply ignored.

As a work-around, to make all of the characters in the sequence (the special character and the digits) appear in the same run, the explicit directional override U+202E RIGHT-TO-LEFT OVERRIDE may be used to make sure that all the following characters fall into a same run. See Figures 2 and 3 for examples.

Figure 2: rendering of character sequence ⟨U+0600, U+0661, U+0661, U+0662⟩ by Pango 1.14.4 using the SIL font Scheherazade

Figure 3: rendering of character sequence ⟨U+202E, U+0600, U+0661, U+0661, U+0662⟩ by Pango 1.14.4 using the SIL font Scheherazade

With the present Bidi category assigned to the characters, a text rendering engine is forced to "special case them to merge into the digit run and totally ignore the bidi-class"[2].

In discussions dated December 2005 on the Unicode Consortium's bidi mailing list which involved members from Microsoft (Paul Nelson) and SIL (Jonathan Kew) who were the first implementors of the characters in a rendering engine (Uniscribe) and a font (Scheherazade and Lateef) to the best of our knowledge, and also officers from the Unicode Consortium (Asmus Freytag and Ken Whistler), the following matters were confirmed:

1. Present implementations are not affected in an undesirable way if the Bidi category of the special characters under discussion changes[3]. Instead, they may be simplified if the change is made.
2. The perfect Bidi category for these characters, if the introduction of a new bidirectional type was possible, would have been something like a Common Number Terminator[4], a bidirectional character type that changes its type to the adjacent numbers, independent of if they are of Bidi category EN or AN. But as that is impossible because of stability guarantees made that mention "The Bidi_Category values will not be further divided."[5]
3. From the available set of Bidi categories, the category AN may be more desired for the characters, as the numbers that appear next to the characters under discussion will very probably be either originally of Bidi category AN or will change their Bidi category to AN as a result of step W2 of the Bidirectional Algorithm.

Based on the above discussion, we believe that the bidirectional category of the following Unicode characters should be changed from "AL" (Right-to-Left Arabic) to "AN" (Arabic Number):

```
U+0600    ARABIC NUMBER SIGN
U+0601    ARABIC SIGN SANAH
U+0602    ARABIC FOOTNOTE MARKER
U+0603    ARABIC SIGN SAFHA
U+06DD    ARABIC END OF AYAH
```

This will ensure that the special characters under discussion will appear in the same bidirectional run as the digits that follow them, and make text rendering engines have an easier time in handling their strange behavior using existing font technology and rendering algorithms. This way, presenting text that includes such characters could be implemented similarly to Arabic and Syriac joining and shaping behavior, which also happens only inside bidirectional runs.

---

2 Quote from Paul Nelson, about the present behavior of Microsoft's Uniscribe engine regarding the special characters under discussion, in a message dated Thu, 1 Dec 2005, 16:49:57 -0500.

3 For example, Paul Nelson, in the same email message mentioned above, mentions that "the implementation we have done to support this functionality in the Windows OS will not change if the bidi class is changed."

4 Suggestions for a new category "AT" (Arabic Terminator) were also made in the discussions.

5 From "Stability Policy for the Unicode Standard", section Property Value Stability, available from http://unicode.org/standard/stability_policy.html#Property_Value

Acknowledgments

The creation of this proposal has been partially sponsored by The Secretariat of the High Council of Information Dissemination of Iran (دبیرخانهٔ شورای عالی اطلاع‌رسانی کشور), through a fund provided to The FarsiWeb Project, Sharif FarsiWeb, Tehran.