

Re: Arabic cursive in Unicode

Source: <http://sci.tech–archive.net/Archive/sci.lang/2006–11/msg01583.html>

- *From:* Mike Wright <news@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 22 Nov 2006 10:38:12 –0600
-

Danny wrote:

Ruud Harmsen wrote:

21 Nov 2006 20:33:38 –0800: "Peter T. Daniels"
<grammatim@xxxxxxxxxxxxx>: in sci.lang:

I mean exactly what I think you think I mean
:) I mean the four (or
two, or one) variants of a letter (or ligature)
that are displayed
according to its position within a word. I'm
pretty sure the term is
the one used by Unicode – I was reminded
to use it by Andreas' post.

I'm sorry, but I have no idea what you mean. "Presentation
form" is not
a term used in the study of writing systems, or of Arabic, or
in
typography.

But it is in Unicode. <http://rudhar.com/lingtics/uniclnks.htm>
0600 is Arabic, FB50 is Arabic Presentation Forms–A and FE70 is Arabic
Presentation Forms–B. The question may be: should these be used, and
if so, how?

As far as I can see, the answer is that the logical (0600) characters
should be used for data storage (eg a digital file) while the
presentation forms should be used for display (print or screen). The
logical characters only have a visual form for convenience.

As you say, the 0600 range of codes represent abstract "characters", but not concrete glyphs. In Arabic, glyph
forms are context–dependent. The "Presentation forms" provide glyphs for pretty much the full variety of
expected contexts.

Re: Arabic cursive in Unicode

Input methods must deal with this. For example, on the Mac, as you type an Arabic character, you see an isolated form or a final form.

If the preceding character is non-alphabetic, you see the isolated form. Otherwise, you see the final form, which connects to the preceding character if appropriate.

When another alphabetic character is typed, the previous final form changes automatically to a non-final form (if such a form exists for that character). The newly-typed character will be a final form.

In TextEdit on the Mac, the glyphs that appear on screen are from the Presentation Forms-B list.

However, when the text is stored as plain text, what is stored is in the 0600 range. This means that the display software has to look at the context of the "logical character codes" and select the appropriate "presentation glyphs" every time the text has to be re-displayed.

It would be up to the software whether to give the user the option of displaying the ligatures from the Presentation Forms-A list. I suspect that this kind of option would only be available in specialized Arabic-language input software. On the Mac, the Character Palette can be used to input glyphs.

It's worth mentioning that not all fonts handle the Presentation Forms-A ligature glyphs as I would have expected. For example, Unicode character FC0B "Arabic Ligature Teh With Jeem Isolated Form" shows the "Teh" inverted above the "Jeem" in the Al Bayan font family, but not in the Geeza Pro family, which shows the same thing you'd get from the Teh-Jeem combination in Presentation Forms-B.

If you are dealing with keyboard input that is not handled by your OS, what you probably need is a set of context rules for converting a string of 0600-range codes into FE70-range codes for display. If you are just storing unmodifiable strings of canned text for display, then you could probably store the FE70-range codes, once you figure out which ones are appropriate.

Where is your Arabic text coming from in the first place? You say elsewhere: "It wouldn't be practical to get a native speaker of every language we want to support." How much text are you talking about?

--

Mike Wright

<http://www.raccoonbend.com>

.