

Re: Arabic cursive in Unicode

Source: <http://sci.tech-archive.net/Archive/sci.lang/2006-11/msg01527.html>

- *From:* "Peter T. Daniels" <grammatim@xxxxxxxxxxx>
 - *Date:* 21 Nov 2006 13:11:12 -0800
-

Danny wrote:

Andreas Prilop wrote:

On 21 Nov 2006, Danny wrote:

I'm trying to draw up a table of Arabic cursive characters for a text editor: I want to take the raw data and translate into a sequence of cursive variants.

What do you mean by "cursive characters", "cursive variants"?

I mean that I need to take the 'actual' characters (in the 06 range) and convert to the presentation forms (in the F range). This is in order to make my own Arabic rendering engine (bypassing the OS)

What do you mean by "presentation forms"? All the "isolated" shapes of letters can occur within or at the end of words -- the "four" shapes of Arabic letters exist only because some letters can connect with letters on both sides, and some can connect only with letters before them (to their right).

Note that, back in Apple's Worldscript I, which was how you typed Hebrew and Arabic back in System 7, the standard Arabic font (which fitted into 256 characters minus control characters) accommodated Arabic, Persian, and Urdu (certainly the three most used Arabic-written languages) -- but a few of the vowel points were omitted, such as the dagger alif and the wasla. It did the contextual forms automatically and even handled the most common ligatures.

Then someone came up with the predecessor of Open Type, which was called GX, which could do a very good job of imitating a handwritten

Re: Arabic cursive in Unicode

text (so many ligatures included), but the Arabics that come with Windows these days have regressed.

An example: the letter Alef Maksura (0649) exists in an isolated and final form at points FEEF and FEF0, but the initial and medial forms are listed at FBE8 and FBE9 (with the complicated name ARABIC LETTER UIGHUR KAZAKH KIRGHIZ ALEF MAKSURA INITIAL FORM). To me, this suggests that standard Arabic includes only the first two forms, and the other two only appear in a variant.

Never rely on the Unicode *names* for characters! They are never changed and may be misleading. The most prominent example is the "byte order mark" U+FEFF, which will be known forever as "zero-width no-break space". Therefore, do not infer anything from the *name* "alef maksura".

Sure – however, this is also defined as the initial form of character 0649, so the name matches the coding.

Second, do not rely on "compatibility characters" such as "Arabic presentation forms". They exist mainly for compatibility with older character sets. Never use them.

Huh? Now I'm confused. These presentation forms are the actual characters displayed on screen when viewing Arabic, aren't they? If not, where in the font are these presentation forms stored?

Anyway: fortunately in my case I'm including the font outlines in my application, so I can work with the glyphs as they are stored in the font; I don't need to worry about future-proofing.