Die TEXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung TEX e.V.

Idris Samawi Hamid: *Introducing Arabic-Latin Modern Fixed*,
Die TEXnische Komödie 3/2014, S. 19-46.

Introducing Arabic-Latin Modern Fixed

Idris Samawi Hamid

DANTE Summary Report:
Introducing Arabic-Latin Modern Fixed

Idris Samawi Hamid

1 Background: The Oriental TEX Project

The Oriental TEX project was initiated in 2006 to facilitate the development of high quality typography and typesetting of academic and scholarly texts that require the Arabic script, such as critical editions and monographs. Although support for the Arabic script in modern typesetting software has been slowly improving over the past decade or so, the situation is still very far behind the Latin script in terms of features, available high-quality typefaces, and layout-processing software. For academic and scholarly work, it's still very much a wilderness out there. A full solution to the problems of advanced Arabic-script typography and typesetting, particularly one based on OpenType and Unicode standards, is still some ways off.

So far the Oriental TEX project has worked closely with the LuaTEX project, also initiated in 2006; an initial large-sum grant from Colorado State University in particular was an important boost to LuaTEX development. In the last round of supplemental funding from DANTE (2009) we focused on the development, implementation, and testing of the OpenType layout engine in ConTEXt's LuaTEX-based MkIV (in close collaboration with Hans Hagen).[1] In this current round of funding through DANTE, we are focusing on a different aspect of the mission of Oriental TEX, to be described in the next section.

2 The Problem

2.1 Editing Marks in an Abjad Writing System

Unlike Latin script, which is an *alphabetic* writing system, Arabic script is an *abjad* writing system. The key difference between the two is as follows:

An *alphabetic* writing system consists of *letters* that represent consonant sounds, as well as characters that represent vowel sounds. Thus in an alphabetic script each fundamental sound in a given language,[2] whether a consonant (e. g. ›b‹, ›ch‹) or a

[1] See »OpenType Engineering in TEX«, *Die TEXnische Komödie*, 4/2009.

[2] The technical term for such a fundamental sound is ›phoneme‹.

vowel sound (e. g., ›i‹, ›ai‹), is represented by one or more letters of the alphabet of that language.

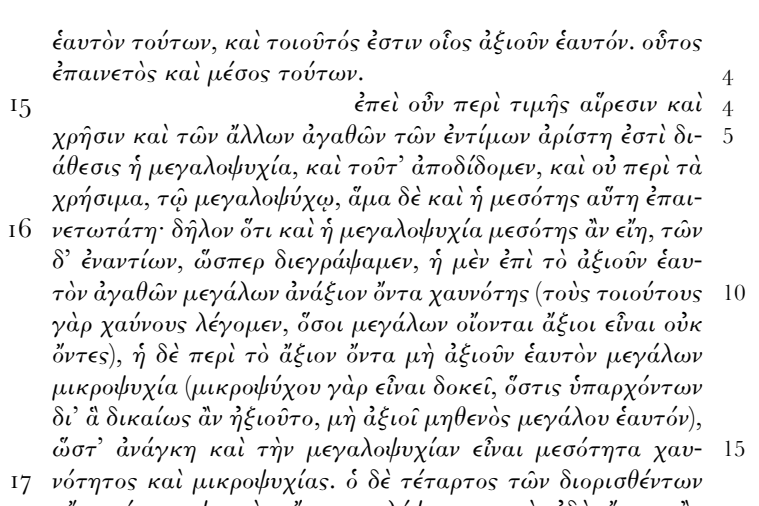
On the other hand an *abjad*[3] writing system is one where each letter represents a consonant sound, and most vowels are not represented by letters. Sometimes a consonant may function as a vowel in certain contexts in addition to its consonantal function (in Arabic the equivalent of the letters ›a‹, ›w‹, and ›y‹ do this), but most vowels remain implicit rather than explicit.

Thus a single letter string can be quite ambiguous and represent a number of different words. Furthermore, in the case of Arabic in particular the grammatical role of words in a sentence are determined by *inflection*, where the last letter of a word takes on a different sound depending on its context. In order to disambiguate words and inflection endings, early scholars of Arabic developed a system of diacritics and vowel markings to mark each pure consonant within a word. This was especially motivated by the case of the Qurʾān; within a generation of the passing of the Prophet of Islam the non-Arabic-speaking peoples who were converting to Islam in large numbers would frequently get especially the inflections incorrect, inadvertently affecting the meaning of Qurʾānic expressions in disastrous ways. For the purpose of precisely presenting a Classical Arabic text in a modern edition, getting the vowel markings correct is also crucial. However, entering the markings correctly for the purposes of typesetting has always been an arduous task. It is instructive to compare this with the situation in scholarly Greek.

Although its script constitutes an alphabetic system (as opposed to an abjad one), the precise representation of many aspirations (or »breathings«) and accents in Ancient and Hellenistic Greek came to be be represented by a sophisticated system of diacritical markings. The entire Greek alphabetic script inclusive of the full system of diacritics is called *polytonic Greek*. In Unicode this system is represented by characters located within the Greek-Extended Block (U+1F00..U+1FFF). Most of these polytonic letters can be composed in Unicode from a combination of basic Greek consonants (located within the U+0370..U+03FF block) and other diacritics (mostly common to both Greek and Latin typography). For example, ›Ἀ‹ (U+1F08) can be decomposed into ›Α‹ and ›᾽‹. This makes the digital entry of polytonic Greek text quite straightforward and easy to edit: One can easily proofread and edit a manuscript either

a. by selecting and modifying each diacritic independent of the letter that it modifies (using decomposition); or
b. by replacing one Unicode code point (e. g. ›Ἀ‹) with another (e. g. ›Ἁ‹).

[3] The term ›abjad‹ comes from the first four letters of most Semitic languages, ›A‹, ›B‹, ›J‹, and ›D‹.


ἑαυτὸν τούτων, καὶ τοιοῦτός ἐστιν οἷος ἀξιοῦν ἑαυτόν. οὗτος
ἐπαινετὸς καὶ μέσος τούτων. 4
15 ἐπεὶ οὖν περὶ τιμῆς αἵρεσιν καὶ 4
χρῆσιν καὶ τῶν ἄλλων ἀγαθῶν τῶν ἐντίμων ἀρίστη ἐστὶ δι- 5
άθεσις ἡ μεγαλοψυχία, καὶ τοῦτ' ἀποδίδομεν, καὶ οὐ περὶ τὰ
χρήσιμα, τῷ μεγαλοψύχῳ, ἅμα δὲ καὶ ἡ μεσότης αὕτη ἐπαι-
16 νετωτάτη· δῆλον ὅτι καὶ ἡ μεγαλοψυχία μεσότης ἂν εἴη, τῶν
δ' ἐναντίων, ὥσπερ διεγράψαμεν, ἡ μὲν ἐπὶ τὸ ἀξιοῦν ἑαυ-
τὸν ἀγαθῶν μεγάλων ἀνάξιον ὄντα χαυνότης (τοὺς τοιούτους 10
γὰρ χαύνους λέγομεν, ὅσοι μεγάλων οἴονται ἄξιοι εἶναι οὐκ
ὄντες), ἡ δὲ περὶ τὸ ἄξιον ὄντα μὴ ἀξιοῦν ἑαυτὸν μεγάλων
μικροψυχία (μικροψύχου γὰρ εἶναι δοκεῖ, ὅστις ὑπαρχόντων
δι' ἃ δικαίως ἂν ἠξιοῦτο, μὴ ἀξιοῖ μηθενὸς μεγάλου ἑαυτόν),
ὥστ' ἀνάγκη καὶ τὴν μεγαλοψυχίαν εἶναι μεσότητα χαυ- 15
17 νότητος καὶ μικροψυχίας. ὁ δὲ τέταρτος τῶν διορισθέντων
οὔτε πάμπαν ψεκτὸς οὔτε μεγαλόψυχος, περὶ οὐδὲν ἔχ‹ον› ὧν
μέγεθος· οὔτε γὰρ ἄξιος οὔτε ἀξιοῖ μεγάλων, διὸ οὐκ ἐναν-
18 τίος. καίτοι δόξειεν ἂν ἐναντίον εἶναι τῷ μεγάλων ἀξίῳ ὄντι
μεγάλων τὸ μικρῶν ὄντα ἄξιον ‹μικρῶν› ἀξιοῦν ἑαυτόν. οὐκ 20
ἔστι δ' ἐναντίος οὔτε τῷ μὴ

1233[a] 9–30 = *EN* 1125[a]16–34; cf. 1122[b]30–1123[a]18

3 *οἷον ἀξιοῖ ἑαυτόν·* Richards: *οἷος ἀξιοῦν ἑαυτόν*, codd. 6 ‹*περὶ*› suppl. Walzer 7 *τὰ* post *οὐ* suppl. Fritzsche, *τὸ* Ross *τὸν μεγαλόψυχον* codd.: *τῷ μεγαλοψύχῳ* Fritzsche 8 *αὕτη* secl. Rackham 10 *ἐπὶ* codd. (cf. 22[a]23): *ἐπεὶ* Ald.: *περὶ* Ross (cl. 33[a]12) *τὸ*] *τῷ* L

Figure 1 Polytonic Greek in action (image provided by Thomas Schmitz).

Classical editions take the effort to properly represent polytonic Greek in all its glory (see Figure 1). Monotonic Greek (i. e., Greek script mostly stripped of the diacritics) is thus hardly a credible option in Classical scholarship, philology, or textual studies (except perhaps for certain narrow, specialized cases).

Unfortunately, in the Arabic script and language the situation is much more complicated. Not only is there an array of intra-word vowels plus inflections that have to represented on most letters (by *marks*), but each letter can take on a multiple of shapes, with almost limitless possibilities. Representing each and every one of these possible letter-mark combinations by a single Unicode code point is unwieldy and impractical. A quick look at the (in my strong opinion) very ill-conceived Arabic Presentation Forms A (U+FB50..U+FDFF block) will let one view a non-exhaustive list of some possible letter-shape possibilities that is virtually unworkable for digital typesetting purposes. Now if one were to add all of the possible letter-mark combinations for each character of the two Presentation-Forms blocks (A and B,

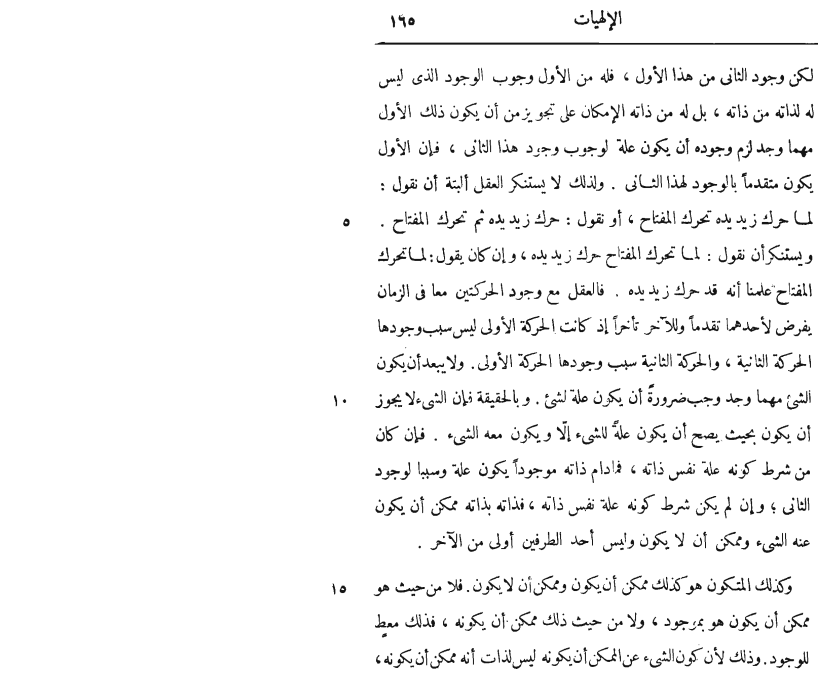
for a total of 891 characters) the number of resultant code points would expand exponentially.

Even if we consider only the basic four Arabic-script shapes-per-letter needed for normal text entry into an editor (isolated, initial, medial, and final), managing separate code points for each possibility still becomes quite impractical. Hence the OpenType solution of entering marks separate from base letters, then combining them using OpenType rules which are then displayed by the layout engine. That layout engine may be within a text editor or a text processor such as TEX.

But this is where the problem begins. OpenType rules specify *general positioning* (GPOS) rules for the *vertical* placement of marks over base letters. This is entirely correct and indispensable (see Figure 2). On the other hand, in a text editor this makes it very impractical and tiresome to proofread and correct vowels in an Arabic text of any substantial length. This is because in an editor one selects text, not vertically, but *horizontally*. Anyone who has tried to select and edit Arabic-script marks in an editor separately from the consonants knows what a pain it is. It is a two-dimensional task boxed into a one-dimensional standard of digital input with a cursor. No text editor or processor of which I am aware has ever been designed to handle two-dimensional input.

Figure 2 Mark on consonant in Arabic.

The above problem is not unique to digital typography. Lead-press typography as well had a hard time dealing with marked Arabic script. Ironically, the best examples of Arabic-script typesetting with marks over multiple shapes is found in early 20^{th} century Arabic typography. But over the following decades the use of marks decreases, along with less and less shape variations of the letters themselves. This is due, no doubt, to the overall unwieldiness and growing expense of the process. So Arabic critical editions are rarely, if ever, done to the same degree of orthographical precision as a Classical-Greek edition (see Figure 3 for a standard example of an Arabic critical edition).



١٦٥ الإلهيات

لكن وجود الثانى من هذا الأول ، فله من الأول وجوب الوجود الذى ليس له لذاته من ذاته ، بل له من ذاته الإمكان على تجويز من أن يكون ذلك الأول مهما وجد لزم وجوده أن يكون علة لوجوب وجود هذا الثانى ، فإن الأول يكون متقدماً بالوجود لهذا الثانى . ولذلك لا يستنكر العقل ألبتة أن نقول : لما حرك زيد يده تحرك المفتاح ، أو نقول : حرك زيد يده ثم تحرك المفتاح . ويستنكر أن نقول : لما تحرك المفتاح حرك زيد يده ، وإن كان يقول : لما تحرك المفتاح علمنا أنه قد حرك زيد يده . فالعقل مع وجود الحركتين معا فى الزمان يفرض لأحدهما تقدماً وللآخر تأخراً إذ كانت الحركة الأولى ليس سبب وجودها الحركة الثانية ، والحركة الثانية سبب وجودها الحركة الأولى . ولا يبعد أن يكون الشئ مهما وجد وجب ضرورةً أن يكون علة لشئ . وبالحقيقة فإن الشىء لا يجوز أن يكون بحيث يصح أن يكون علةً للشىء إلّا و يكون معه الشىء . فإن كان من شرط كونه علة نفس ذاته ، فمادام ذاته موجوداً يكون علة وسببا لوجود الثانى ؛ وإن لم يكن شرط كونه علة نفس ذاته ، فذاته بذاته ممكن أن يكون عنه الشىء وممكن أن لا يكون وليس أحد الطرفين أولى من الآخر .

وكذلك المتكون هو كذلك ممكن أن يكون وممكن أن لا يكون . فلا من حيث هو ممكن أن يكون هو بموجود ، ولا من حيث ذلك ممكن أن يكونه ، فذلك معطٍ للوجود . وذلك لأن كون الشىء عن الممكن أن يكونه ليس لذات أنه ممكن أن يكونه ،

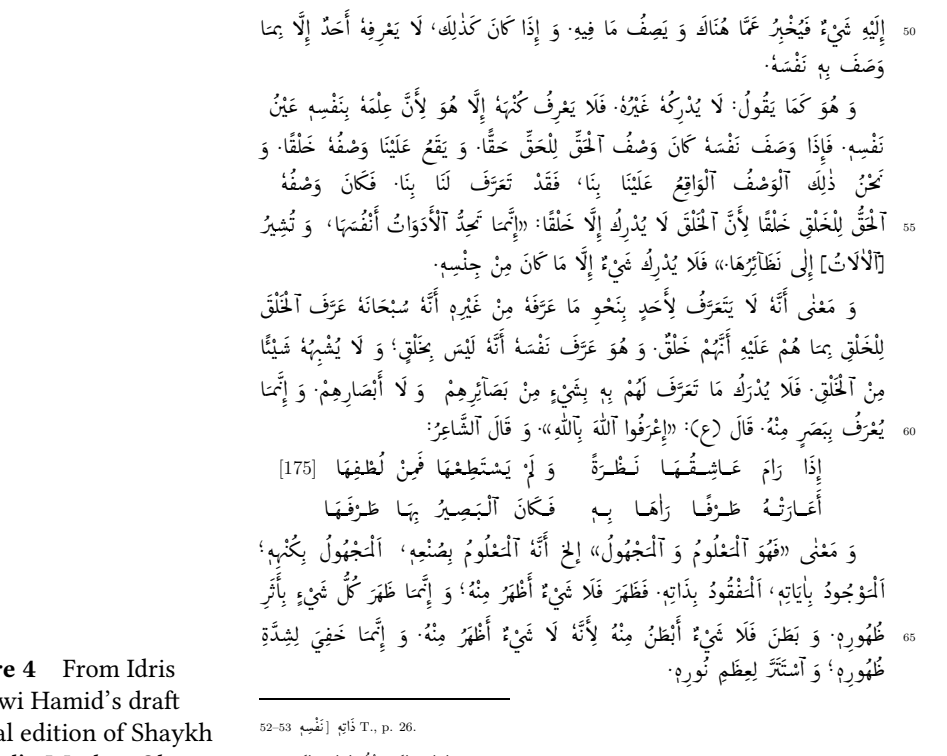
(١) وجود : + الشئ ب ج ، د ، ص ، ط (٤) ولذلك : ولهذا م (٥) تحرك المفتاح أو تقول .... يده : ساقطة من م (٧) الزمان : زمان د ، ط (٨) إذ : إذا د ، ص ؛ إذ لو ط || الحركة : حركة د || ليس : ليست ج ، ص (١٠) الشئ : الشئ ب ج ، د ، ص ، ط || وبالحقيقة : بالحقيقة ط ، م (١١) الشئ : + وبالحقيقة ب (١٢) من : ساقطة من ب ، م (١٣) فذاته بذاته : ساقطة من د || ممكن : يمكن د (١٤) الطرفين : + له ج ؛ + به ص (١٥) وكذلك : فكذلك د ، ط ، م || المتكون : المكون ط ؛ المتكرد (١٦) بموجود : موجود ب || ممكن أن يكونه : أن يكونه ممكن ط (١٧) كون : يكون ط .

Figure 3 From George Anawati's critical edition of Ibn Sīnā's *Metaphysics* (1950's).

As a part of my doctoral dissertation I did a critical edition of the Arabic text of *Wisdom Observations* by Shaykh Aḥmad al-Aḥsā'ī (d. 1826). In this case full marks were applied (see Figure 4), a task made immensely easier by the use of Klaus Lagally's ArabTEX transcription method. This method involves an entirely romanized Latin transcription of the Arabic script, using combinations of one or more Latin vowel letters to represent Arabic marks. If this had to be done in a text editor with a digital Arabic font the task of managing the marks would have been much more difficult.

With the growing popularity of Unicode and its implementation within TEX (particularly XETEX and CONTEXT) Lagally's method has become mostly obsolete. Thankfully, today no one should have to enter text via Lagally's transcription method (as complete and as reliable as it was), particularly for extensive texts. Yet the problem

of entering and editing marks within digital Arabic text remains a thorn in the side of scholars who have to work with the Arabic script on a continual basis.



427

إِلَيْهِ شَيْءٌ فَيُخْبِرُ عَمَّا هُنَاكَ وَ يَصِفُ مَا فِيهِ. وَ إِذَا كَانَ كَذٰلِكَ، لَا يَعْرِفُهُ أَحَدٌ إِلَّا بِمَا وَصَفَ بِهِ نَفْسَهُ.

وَ هُوَ كَمَا يَقُولُ: لَا يُدْرِكُهُ غَيْرُهُ. فَلَا يَعْرِفُ كُنْهَهُ إِلَّا هُوَ لِأَنَّ عِلْمَهُ بِنَفْسِهِ عَيْنُ نَفْسِهِ. فَإِذَا وَصَفَ نَفْسَهُ كَانَ وَصْفُ ٱلْحَقِّ لِلْحَقِّ حَقًّا. وَ يَقَعُ عَلَيْنَا وَصْفُهُ خَلْقًا. وَ نَحْنُ ذٰلِكَ ٱلْوَصْفُ ٱلْوَاقِعُ عَلَيْنَا بِنَا، فَقَدْ تَعَرَّفَ لَنَا بِنَا. فَكَانَ وَصْفُهُ ٱلْحَقُّ لِلْخَلْقِ خَلْقًا لِأَنَّ ٱلْخَلْقَ لَا يُدْرِكُ إِلَّا خَلْقًا: «إِنَّمَا تَحِدُّ ٱلْأَدَوَاتُ أَنْفُسَهَا، وَ تُشِيرُ [ٱلْآلَاتُ] إِلَى نَظَآئِرِهَا.» فَلَا يُدْرِكُ شَيْءٌ إِلَّا مَا كَانَ مِنْ جِنْسِهِ.

وَ مَعْنَى أَنَّهُ لَا يَتَعَرَّفُ لِأَحَدٍ بِنَحْوِ مَا عَرَّفَهُ مِنْ غَيْرِهِ أَنَّهُ سُبْحَانَهُ عَرَّفَ ٱلْخَلْقَ لِلْخَلْقِ بِمَا هُمْ عَلَيْهِ أَنَّهُمْ خَلْقٌ. وَ هُوَ عَرَّفَ نَفْسَهُ أَنَّهُ لَيْسَ بِخَلْقٍ؛ وَ لَا يُشْبِهُهُ شَيْئًا مِنْ ٱلْخَلْقِ. فَلَا يُدْرِكُ مَا تَعَرَّفَ لَهُمْ بِهِ بِشَيْءٍ مِنْ بَصَآئِرِهِمْ وَ لَا أَبْصَارِهِمْ. وَ إِنَّمَا يُعْرَفُ بِبَصَرٍ مِنْهُ. قَالَ (ع): «إِعْرَفُوا ٱللّٰهَ بِٱللّٰهِ». وَ قَالَ ٱلشَّاعِرُ:

إِذَا رَامَ عَاشِقُهَا نَظْرَةً وَ لَمْ يَسْتَطِعْهَا فَمِنْ لُطْفِهَا [175]
أَعَارَتْهُ طَرْفًا رَأَهَا بِهِ فَكَانَ ٱلْبَصِيرُ بِهَا طَرْفَهَا

وَ مَعْنَى «فَهُوَ ٱلْمَعْلُومُ وَ ٱلْمَجْهُولُ» إِلخ أَنَّهُ ٱلْمَعْلُومُ بِصُنْعِهِ، ٱلْمَجْهُولُ بِكُنْهِهِ؛ ٱلْمَوْجُودُ بِآيَاتِهِ، ٱلْمَفْقُودُ بِذَاتِهِ. فَظَهَرَ فَلَا شَيْءٌ أَظْهَرُ مِنْهُ؛ وَ إِنَّمَا ظَهَرَ كُلُّ شَيْءٍ بِأَثَرِ ظُهُورِهِ. وَ بَطَنَ فَلَا شَيْءٌ أَبْطَنُ مِنْهُ لِأَنَّهُ لَا شَيْءٌ أَظْهَرُ مِنْهُ. وَ إِنَّمَا خَفِيَ لِشِدَّةِ ظُهُورِهِ؛ وَ ٱسْتَتَرَ لِعِظَمِ نُورِهِ.

52–53 ذَاتِهِ [نَفْسِهِ T., p. 26.

54–55 لِلْخَلْقِ خَلْقًا [ٱلْحَقُّ لِلْخَلْقِ خَلْقًا T., p. 26.

55 وَ تُشِيرُ ٱلْآلَاتُ [وَ تُشِيرُ T., p. 26.

59 وَ لَا مِنْ [وَ لَا T., p. 27.

63 وَ ٱلْمَجْهُولُ [ٱلْمَجْهُولُ T., p. 28.

Figure 4 From Idris Samawi Hamid's draft critical edition of Shaykh Aḥmad's *Wisdom Observations* (doctoral dissertation; note the extensive use of marks).

2.2 Enter SC Unipad

For some time this author struggled over the choice between the use of Unicode-based Arabic script and that of Lagally's romanized transcription. Finally, I discovered a Unicode editor called SC Unipad, a $200.00 utility developed by some Persian-American developers. In addition to having perhaps the best implementation of the Unicode Bidirectional Algorithm in any text editor, the developers implemented a simple yet ingenious solution to the problem of entering text with marks. If the mark belongs to a given letter that is initial or medial, extend the

letter via *taṭwīl* and place the mark *horizontally* to the left of that letter. Consider the following Unicode Arabic text rendered with a simple and basic OpenType font such as SIL's Scheherazade:

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ ۝١

Note the vertical marks above and below the base letters. Now let's compare the above to precisely the same string rendered by Unipad (see Figure 5). Note the positioning of the marks now makes them easily editable in one dimension (horizontal).

As exciting as this development was (and Unipad has lots of useful features for editing Unicode text) it soon became clear that this was a solution with many limitations. For example:

- The interline spacing is too tight, so marks come very close to clashing.
- The Unicode font is bitmap and hardcoded into the application. This means it cannot be used in any other editor.
- Although designed for its size (making it crystal clear at that size) it cannot be enlarged or shrunk.
- A TEX-code editor needs lots of important features if one is to have an efficient workflow; Unipad is hardly any good at all as a TEX-code editor.
- Perhaps worst of all, Unipad hasn't been updated since 2006 and is only up to Unicode version 4.1.0 in compatibility (Unicode is currently at 7.0). Numerous characters (including from Arabic script) are thus missing.

Despite its severe limitations, SC Unipad had indeed found a simple and ingenious solution to the problem of entering and editing Arabic-script marked text. The more general solution to the problem, then, would be to develop an OpenType font with capabilities similar to (and beyond) those of Unipad's bitmap font.

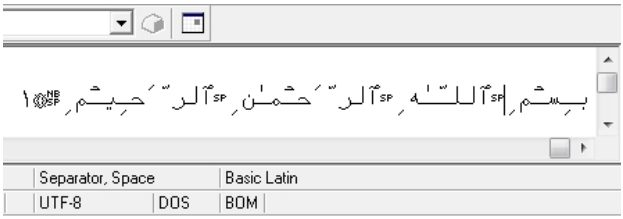


Figure 5 SC Unipad and Arabic-script marks.

2.3 Verbatim

In the process of consulting with Hans Hagen in the ongoing development of ConTeXt's bidirectional capabilities, lots of verbatim issues arise. Verbatim is important because it helps to illustrate how a string of text is supposed to behave at different levels of the bi-directional analysis. So a common procedure would be to place a string of text into a buffer, then process that buffer to show both verbatim as well as fully processed output in the resultant pdf. No current font really meets the needs for verbatim Arabic text. So far we have been using DejaVu Sans Mono, whose limitations and lacunae as far as Arabic script are concerned are considerable.

A verbatim font is usually a fixed-width text font, the same kind of font generally used in a text editor. Thus a properly designed OpenType fixed-width font would address both issues: that of text editing as well as of verbatim representation in TeX processing.

2.4 Completeness

Finally, there is no generally available fixed-width (or even variable-width) font that is *complete* with respect to Arabic script. Nearly every available text-editing font is missing important characters. Any solution to the above problem would need to be as complete as possible as well as support the Arabic-script ranges of Unicode 7.0, the latest version as of this writing.

3 From Latin Modern Mono to Arabic-Latin Modern Fixed

3.1 TeX Gyre Latin Modern Mono

For our project we decided to extend Latin Modern Mono Regular 10 to Arabic. We chose this particular font for a number of reasons, including

- Most TeX verbatim work is done in Latin Modern Mono Regular.
- The TeX Gyre fonts are OpenType, and currently support everything in Latin needed for the *transliteration* of Arabic script into Latin. This is crucial for the editing of academic texts that work with transliteration.

TeX Gyre Cursor (close in design style to Courier New, a very nice editing fixed-width font) was also considered as a starting point. On the other hand, the metric width of Cursor is 600 font units, whereas that of Latin Modern Mono is 525 font units. Since the horizontal moving of marks is already going to stretch out the length of word strings, and the vertical metrics are going to have to be increased as well, using a smaller metric width allows a more economical use of space for marked Arabic-script expressions.

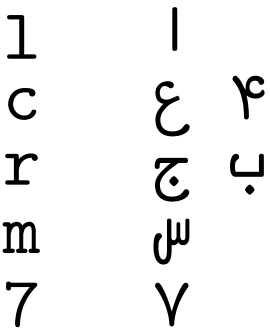
Another reason for going with Latin Modern is in honor of Donald Knuth's original verbatim design (Computer Modern Mono). My own motivations for getting into digital typography in the first place are much the same as those of Knuth, so an extension of Knuth in this instance seemed to be in the spirit of things.

3.2 Design

3.2.1 Knuthian Inspiration

The Arabic-script portion of Arabic-Latin Modern Fixed is a completely original font design, not based on any other Arabic-script font or typeface. It is inspired in part by the SC Unipad bitmap font. Its developers did an excellent job of designing it for readability at its intended resolution (which is where bitmaps are supposed to excel). But the actual letters of our font were developed from a careful study of Knuth's letters in Latin Modern Mono, such as the characters ›l‹, ›r‹, ›c‹, and ›7‹. The aim was to use elements in the original Knuthian design to develop a culturally authentic and aesthetically pleasing Arabic-script fixed-width font. In other words, Arabic-Latin Modern Fixed is a natural extension of and complement to the Knuthian design and inspiration.

In the following example, the Arabic-script letters are on the right:



Thus one notices that the »ear« of the ›r‹ (top-right »terminal« or extremity) harmonizes well with many of the extremities of the Arabic letters. Now traditional Arabic-script does not contain these »ears«: We integrated the ears into the Arabic design in a way that we hope expands the possibilities of culturally authentic Arabic in a natural way. For example: In traditional Arabic terminal endings in characters are usually tapered from thick to thin. In some of the characters above one will notice we did exactly the opposite: Tapering goes from thin to thick (at the »ears«). Yet I believe we have managed to maintain a powerful, authentic Arabic feel, that is in harmony with the Knuthian design.

3.2.2 Vertical Metrics and Latin

Note that the width and length of the stem of the Arabic letter Alif (ا, first letter of the Arabic alphabet) is exactly the same as that of the Latin lower case Ell (l). Using this as a benchmark, it becomes easy to see why Arabic script nearly always needs more vertical-metric space than Latin in mixed script contexts. The maximum ascenders and descenders for Arabic-Latin Modern Fixed are considerably greater than those of Latin Modern Mono. This is to be expected: As we design the rest of the Arabic letters with respect to Alif, the ascender-descender space naturally has to expand to accommodate things. Note also that the baseline for Arabic letters is higher than it is for Latin: This is also natural and to be expected.

Virtually every aspect of the base Latin Modern Mono design has been incorporated into Arabic-Latin Modern Fixed. I did notice defects in a number of glyphs but except in a few minor cases have not changed anything in Latin Modern. The OpenType tables have been ported over. The only thing that prevents Arabic-Latin Modern Fixed from serving as a drop-in replacement for its Latin Modern Mono base is that the vertical metrics are different. Indeed, this is one reason for choosing to name the font ›Arabic-Latin Modern Fixed‹ instead of ›Latin-Arabic Modern‹. Of course, in TEX one can set the interline spacing of Arabic-Latin Modern Fixed to precisely match that of its Latin Modern counterpart.

3.2.3 On-screen Applications

Due to its intended use for editors, a considerable effort has been made to make Arabic-Latin Modern Fixed comfortable for on-screen viewing. A number of Arabic-script characters have gone through numerous iterations in pursuit of this goal. Each shape iteration often involves having to then adjust dozens of other characters that depend on that shape, making this a very time-consuming process (the letter س has been a real challenge, with its very close teeth). I did not change the Latin Modern counterparts (except in some very minor places as mentioned above). On the other hand, the hinting of Arabic-Latin Modern Fixed is, I hold, much better than that of Latin Modern Mono.

One reason I did not previously use Latin Modern Mono for on-screen applications such as text editing is because of its very limited hinting. For example: An on-screen string of, e. g., all capital letters at normal typing sizes (9-14 points) shows sometimes wild inconsistencies in the letter heights. Now, in Arabic-Latin Modern Fixed, the Latin component of Arabic-Latin Modern Fixed looks much better on-screen than in the former; it is to my eye virtually as comfortable as, say, Courier New on Windows.

Arabic-script fonts generally require greater point sizes for readability than Latin. Arabic-Latin Modern Fixed is just readable on-screen with marks even as low as

seven points. Comfortable point sizes for editing purposes range from 9 to 14 points (I particularly like the hinting effects in sizes 11 and 13; your own mileage may vary of course).

3.2.4 Some Notes:

- In a future maintenance release I may fix some of the minor errors in a few of the Latin Modern Mono glyphs.
- Although it has plenty of room for improvement (as almost any font does) the shapes have been more successful than I expected. I can envision Arabic-script applications where this could be used apart from editing or verbatim (such as web-page design).
- Related to this, I may eventually add a proportional version of this design to complement TEX Gyre Latin Modern Proportional 10. Much of the current design elements can be reused. Although not as useful for verbatim and editing purposes, the aesthetics appear to me as robust enough to make a future proportional version a worthwhile pursuit.

3.3 Character Coverage

3.3.1 Arabic Blocks

The original Latin Modern Mono contains 785 glyphs, including about 617 standard Unicode characters. Oriental-TEX Arabic-Latin Modern Mono contains 2630 glyphs, for an increase of 1845 glyphs. Of these, 1113 are Unicode code points. Arabic-Latin Modern Fixed covers every single character in all Unicode-Arabic blocks: Arabic, Arabic Supplement, Arabic Extended, Arabic Presentation-Forms A, and Arabic Presentation-Forms B.

One of the main features of Arabic-Latin Modern Fixed is its unique treatment of Arabic-script marks. Our font contains a total of 66 Unicode marks, more than any other publicly available font of which I am aware. It is almost certainly the most complete fixed-width font as far as the Arabic-script blocks are concerned.[4]

3.3.2 Presentation Forms

The Arabic Presentation Forms A includes what are commonly called »ligatures«, although the ligature concept doesn't really make sense for the Arabic script (for reasons discussed elsewhere). These characters have been decomposed into their base components: For two-component forms the metric width has been doubled; for three-component forms the metric width has been tripled. In part for this reason we call this font »fixed-width« rather than »mono«. That is, in a fixed-width

[4] Nor have I seen any commercial font that covers these ranges so completely. On the other hand, this is not to deny that others may exist or be under development.

font, occasional or special characters may have a width that is a natural-number multiple of the fixed width; in a monospaced font every glyph must have precisely the same width. The expressions ›fixed-width‹ and ›monospaced‹ are normally used synonymously; we have introduced a subtle distinction in our adaptation of this nomenclature. Some examples:

ﺋـﻪ ﻃﺢ ﺳﻢ ﺗـﺠﻢ ﻳـﻤﻢ

Column U+FDFX of Arabic Presentation-Forms A has been maintained without decomposition: A special character has been designed for each, within the standard fixed-width:

ﷺ ﻗﻠﮯ ﷲ اﮐﺒﺮ ﻣُﺤَﻤَّﺪ ﺻﻠﻌﻢ رسول

﷽ ﻋﻠﻴﻪ وَسَلَّمَ صَلَّى ﷺ ﺟَلَّ جلاله ريال

The famous Lām-Alif »ligature« (ﻻ ﻼ) has been maintained *only* for U+FEFB and U+FEFC. In every other case (U+FEF5..U+FEFA) they have been decomposed. In addition, U+0644 plus U+0621 do *not* automatically combine into the Lām-Alif combination. This is because Arabic-Latin Modern Fixed's purpose is primarily to serve as an input, editing, and verbatim font. Ligatures and the like get in the way of that task. Once input is done the text can processed via a layout processor such as TEX to another font (or one can make a simple font switch in a word processor). U+FEFB and U+FEFC (initial and final forms) have been kept because these two have an ancient pedigree in the Arabic script and deserve an independent representation.[5]

With a few exceptions (such as U+FD3E–U+FD3F ﴾ ﴿ ornate parenthesis), the bulk of Presentation-Forms A are useless for digital text entry and their use should be severely discouraged. They are there purely for esoteric legacy purposes, and in my view even those legacy purposes are suspect. Very few applications ever used the bulk of these characters; it is better to convert what few (if any) texts encoded using them into what even the Unicode Consortium calls the »preferred« Arabic Block (0600-06FF). On the other hand, it is the aim of Arabic-Latin Modern Fixed to be complete, so we have included the entire range of presentation characters.

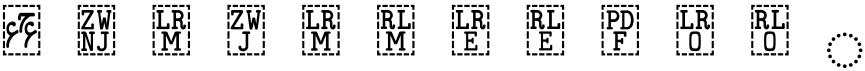
Note that Arabic-Latin Modern Fixed does *not* support initial, medial, or final processing of Presentation Forms; this is intentional. The most these forms should

[5] Indeed, in some ancient texts (ﻻ or U+FEFB) is considered a distinct letter of the Arabic alphabet. On the other hand, U+FEFB proper should only be used in special cases, otherwise U+0644 plus U+0621 should always be input explicitly.

be used for is visual presentation for purposes of conversion to the main Unicode Arabic block.[6]

3.3.3 Format and Spacing Glyph Representations

We added twelve additional Unicode format and spacing characters (with explicit glyph representations) and also designed a glyph representation for Latin Modern U+00A0 (no-break space), as well as U+25CC (dotted circle):



The glyph representations of these characters are important for at least a couple of reasons:

1. For verbatim representation of multi-directional and multilingual text.
2. For visual control and confirmation in text editors.[7]

The Arabic Language Mark —  or U+061C — is a new format character, placed in the main Arabic block. We will discuss its role in bi-directional typesetting in future research. Given the decision by the Unicode Consortium to place it in the Arabic Block, I decided to give it an Arabic-script representation. The abbreviation is short for علامة الحرف العربي, which means »Arabic Language Mark«.

The dotted circle is needed for Uniscribe to implement its »invalid marks« algorithm as well as the individual representation of marks in running text. We will say more about Uniscribe later.

4 OpenType Features

4.1 TEX Gyre Latin Modern Mono and OpenType

As mentioned earlier, Arabic-Latin Modern Fixed has imported all TEX Gyre Latin Modern Mono OpenType features. A few redundancies have been removed, and

[6] Indeed, I strongly considered a purely *implicit* representation of the Presentation Forms in terms of OpenType lookups and the ccmp feature. Uniscribe, Microsoft's ubiquitous OpenType and language-layout processor, would not apply the feature, although the lookups were technically correct (we will discuss some of Uniscribe's other peculiarities as well later in this report). After many experiments and careful analysis it was decided to go ahead with an *explicit* representation of each and every Presentation-Forms character.

[7] Oddly enough, Microsoft Notepad is one of the only text editors that actually let's you see the glyph representations of format (or »control«) and spacing characters. The glyph representation of format characters can also be turned on and off, while those of spacing characters seem to remain on permanently if the font has them.

one important bug that caused Uniscribe-based applications to switch to another fallback font under some conditions has been fixed.

4.2 Scripts and Language-Specific Discretionaries

There are now two supported scripts: Arabic (OpenType tag <arab>) and Latin (tag <latn>). In ConTeXt one activates them using the <script> key (e. g., script=arab).

Given an OpenType script one can implement lookup and feature rules for each language supported by that script. For most purposes languages follow the same script rules so a Default language (OpenType tag <dflt>)[8] is usually enough. In some cases, however, certain modifications are needed. In the case of TeX Gyre Latin Modern Mono, a few conventions related to punctuation and numerals are implemented for a small handful of Latin-script languages.

In a similar spirit we have implemented some lookups for punctuation and numeric conventions for the three primary and most used Arabic-script languages: Arabic, Persian, and Urdu. For each supported language these are implemented under the Discretionary Ligatures feature (<dlig>). Most other Arabic-script languages can use either one of these three conventions or the default language option. When and where necessary, more language-specific discretions (appropriate to the purpose of this font) can be added as needed by users in the future.

I'll give just one example. Persian-Indo numerals (column U+06FX) and Arabic numerals (column U+066X) mostly look the same, but they follow different bidirectional rules.[9] Urdu numerals also look mostly the same. But two or three numerals use a language-specific shape. For example, the numeral ›4‹ takes on a distinctly different shape in Arabic, Persian, and Urdu respectively:

٤ ۴ ۴

Users of each language (or a regionally related one) can thus choose which language to activate via the <locl> feature.

[8] Internally this is converted to DefaultLangSys internally within the OpenType font file, but higher-level syntax usually uses <dflt>.

[9] Loosely, the Arabic numerals have stronger bidirectional characteristics than the Persian-Indo numerals, which basically follow the same rules as Latin-script digits.

Here is a map of currently supported scripts and languages:

- Arabic script <arab>
  - Default <dflt>
  - Arabic <ARA>
  - Persian <FAR>
  - Urdu <URD>
- Latin script <latn>
  - Default <dflt>
  - Azeri <AZE>
  - Crimean Tatar <CRT>
  - Moldavian <MOL>
  - Romanian <ROM>
  - Turkish <TRK>

Latin-script languages support the official features Discretionary Ligatures <dlig>, Fractions <frac>, Lining Figures <lnum>, Oldstyle Figures <onum>, and Slashed Zero <zero>.

In each Arabic-script language (*except* Default) the language-specific features are found under the feature Localized Forms <locl>. For each Arabic-script language (*including* Default) punctuation alternates are found under Discretionary Ligatures <dlig>. Thus all three languages and default (<dflt>) support switching punctuation from Latin to a more authentic Arabic-script look. For example, the period on the left is the usual U+002E, the one on the right is more suited to Arabic-script but does not involve a different Unicode symbol:

. ٠

Notice that the Arabic-script period is both higher (the Arabic-script baseline is also higher) and shaped more squarish. This feature is turned on under <dlig>

Under Localized Forms locl we have situated figure substitutions (discussed above), and even some analogous language-specific letter substitutions such as

ڧ -> ڤ (Persian)

ڡ -> ڤ (Urdu)

Although these localized forms are also encoded in Unicode and can be input directly, <locl> allows users of one language system to, when desired, switch these shapes without changing the encoding of the source text.

4.3 Mark Editing

One of the chief motivations for the development of Arabic-Latin Modern Fixed was improved marks handling. Unfortunately Uniscribe, Microsoft's language-layout processor and the most used one in the world by far, has some built-in,

hardcoded rules that will override OpenType instructions as it sees fit. In other words, Uniscribe is not only an OpenType layout processor but in some cases also functions as a syntax enforcer. These restrictions make it difficult for font developers to be creative and develop features useful for, e. g., Arabic script.

For example, one could begin to approach the problem of moving marks to the left of the modified letter as follows: Since the font is already fixed-width, keep the marks at the same width (in most fonts a mark glyph is always zero width); and do not apply anchor attachment. Doing this will show up correctly when proofing and previewing the OpenType tables, but Uniscribe will override it, forcing the mark glyphs to occupy the same width space as the modified letter.[10]

Overcoming this limitation of Uniscribe (as well as a couple of others) required some OpenType trickery. In our case, we first converted the marks to base characters (in OpenType each character has to be labelled as either base, mark, ligature, or component). Once converted to a base character, the mark can then be treated mostly as a base character by Uniscribe. I say »mostly« because Uniscribe also has other, more esoteric, restrictions as well (we'll mention a couple of cases later).

After converting the marks to base characters, we can then make a substitution: For each initial or medial character, we substitute the mark (now converted to base) for itself plus the Arabic-script letter extender character (*taṭwīl* or U+0640). For example:

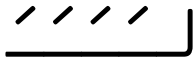
Then we position the mark over the *taṭwīl* using pair-adjustment (GPOS). There are some other subtleties in programming the tables but this is the main idea. So we now have

In a text editor, what had originally been a tedious and impractical task (selecting and editing marks) is just as straightforward as editing any other base character.

[10] Some fonts that come with Windows operating systems (such as Courier New) also have mark glyphs that are not zero-width but fixed width. In either case Uniscribe ignores the width of mark glyphs.

4.4 Esoteric Subtleties

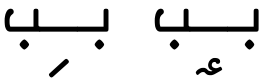
In Uniscribe it is not allowed to form a string of marks in succession, e. g.,



When an »invalid« string is encountered Uniscribe will often place the invalid mark over a dotted circle (◌ or U+25CC). We have added this symbol to Arabic-Latin Modern Fixed. Now, according to Microsoft,[11] »It should also be noted that the dotted circle is not inserted into the application's backing store. This is a run-time insertion into the glyph array…«. This makes it very difficult to override via OpenType trickery, at least without unwanted side effects. However, we did find a workaround: Using the Zero-Width Joiner (ZWJ or U+200D) and contextual substitution of the dotted circle with the *taṭwīl* the following string in a Uniscribe-based application will give the same effect as above – with format-character glyph representation turned *off*, as is default. If you turn the format-character glyph representations *on* you will see something like this in Uniscribe-based applications:

This particular case is somewhat esoteric, but will help make Arabic-Latin Modern Fixed more useful and complete. In ConTeXt this trick is unnecessary.

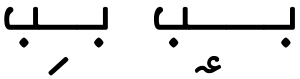
Related: In Uniscribe, newly added marks to the Unicode standard will not be supported unless their status is explicitly hard-coded into Uniscribe. This can be a nightmare for some Arabic-script users of Windows text-processing applications. Consider the following two Arabic-script contextual strings:



Both strings are rendered correctly in ConTeXt. The one on the right uses a mark whose introduction to the Unicode standard is rather recent. Since Uniscribe does not recognize it, even though the OpenType tables characterize it as a mark it doesn't get properly displayed by Uniscribe no matter what. Instead Uniscribe gives

[11] http://www.microsoft.com/typography/OpenTypeDev/arabic/intro.htm

Our OpenType lookups allow a user to partially get around this by adding a *taṭwīl* before the unsupported mark, but that will add an extra *taṭwīl*. For Uniscribe applications this results in



Until there are more readily accessible alternatives to Uniscribe (which is by far the most used and depended-upon multi-lingual OpenType language-layout processor out there) these kinds of issues are things one simply has to live with. Arabic-Latin Modern Fixed has worked around some issues and/or ameliorated others; but given the run-time nature of Uniscribe's limitations OpenType trickery cannot by itself eliminate all peculiarites of Uniscribe. Fortunately, in ConTeXt these issues do not arise, and at least the TeX community is not locked in to the interests or time-table of a commercial corporation to get support as needed.

5 Conclusion

Arabic-Latin Modern Mono will be a boon to scholars, academics, coder editors, and anyone who wants to manage the entry of the wide array of Unicode Arabic text for editing, verbatim, or other purposes. In this regard I believe it is unique (I have not found a single other fixed-width font that comes close to matching it). Arabic-script critical editions can now use the full array of available marks at an expense no greater than that of other characters. I will periodically issue maintenance releases as needed to fix bugs or, on a selective case-by-case basis, add features needed by users (such as other language-dependent discretionaries that may arise).

6 Appendix

6.1 Sample of Arabic-Latin Modern Fixed in ConTeXt

Normally one gives displays a text in verbatim, then gives the result after processing. In this case the main text is the same in both cases (since Arabic-Latin Modern Fixed is also the verbatim font). In addition, getting verbatim bidi correct is for a future project (one that the current font project helps to pave the way for). Here we present the preamble and postscript (without the body explicit, only an indication of the body buffer in its place), then we show the output.

Preamble and postscript:

```
\setupbodyfont[modern,tt]
\definefont[ALM][file:almmono10-regular*arabic at 14pt]
\setuplayout[width=6in]
\setupwhitespace[big]
\starttext
\setupinterlinespace[line=4.9ex] % One can play with this.
\setupalign[r2l]
\ALM

\getbuffer[almmono-sample] % Sample goes in this buffer.

\stoptext
```

The result can be seen in Figure 6.

6.2 License

Upon release, Arabic-Latin Modern Fixed will be released under the GUST Font License or something very close in spirit to it. See

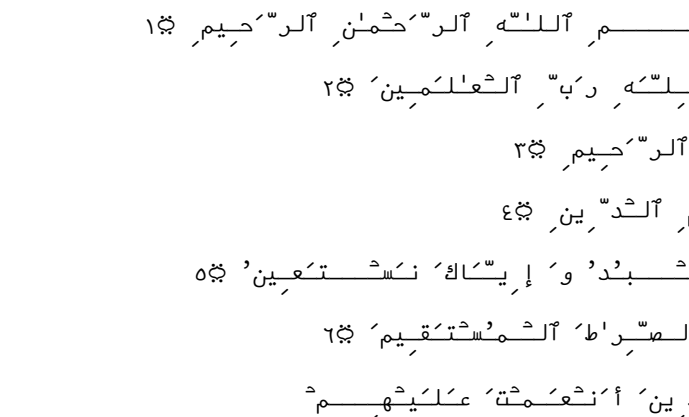
http://www.gust.org.pl/projects/e-foundry/licenses

6.3 Unicode Arabic-Script Blocks

Notes:

- The grayed blocks are *unassigned* Unicode code points (as of version 6.3; see the last bullet); every *assigned* code point is supported by Arabic-Latin Modern Fixed as displayed here.
- The last page of Arabic Presentation Forms-A (FD20-FDFF) uses a smaller font size to fit the larger strings.
- The characters marked in black in the last page of Arabic Presentation Forms-A (FD20-FDFF) are »noncharacters«, i. e., these »codes are intended for process-internal uses« and will never be given character assignments by Unicode.
- Only the primary Unicode code points of Arabic-Latin Modern Fixed are displayed. The various variant and alternate forms are in the remainder of the font.
- As this report was going to press, Unicode 7.0 was released (the previous release was version 6.3). I have gone back and updated Arabic-Latin Modern Fixed to the latest Unicode. The new characters are in the 0600 block (one character) and the 08A0 block (eight characters). You'll notice that each new character is

in a gray box (meaning it was unassigned until the latest version). Again, any variant and alternate forms of these characters will be found in the remainder of the font.



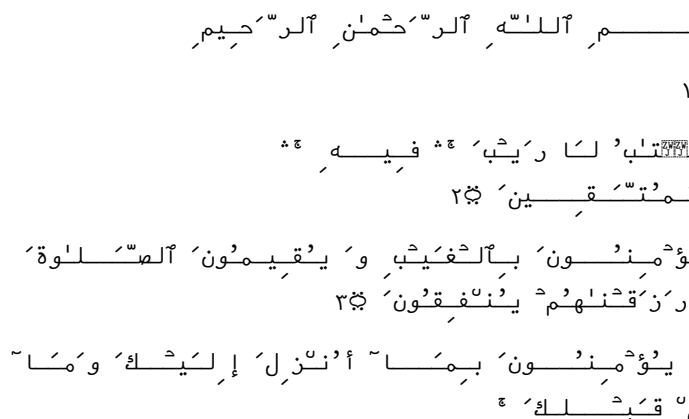


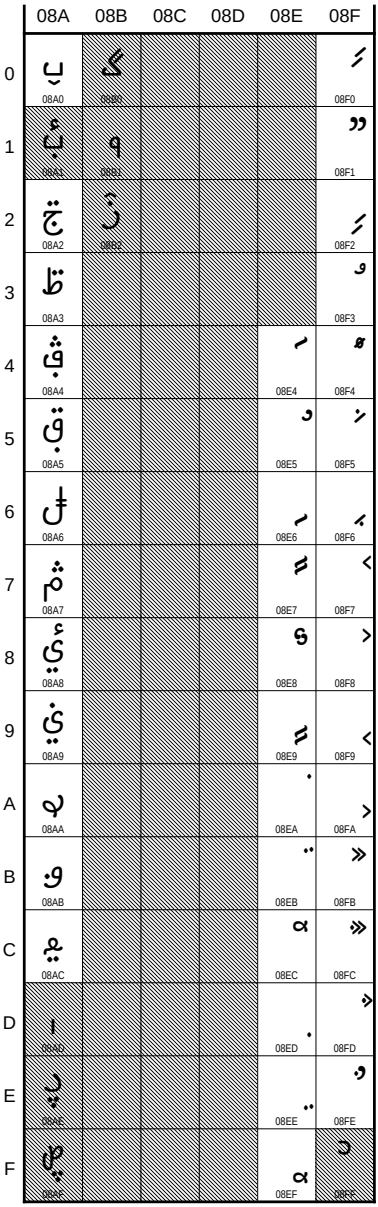
Figure 6 Sample of Arabic-Latin Modern Fixed in action.

0600 **Arabic** **06FF**

0750 **Arabic Supplement** **077F**

	075	076	077
0	ݐ 0750	ݠ 0760	ݰ 0770
1	ݑ 0751	ݡ 0761	ݱ 0771
2	ݒ 0752	ݢ 0762	ݲ 0772
3	ݓ 0753	ݣ 0763	ݳ 0773
4	ݔ 0754	ݤ 0764	ݴ 0774
5	ݕ 0755	ݥ 0765	ݵ 0775
6	ݖ 0756	ݦ 0766	ݶ 0776
7	ݗ 0757	ݧ 0767	ݷ 0777
8	ݘ 0758	ݨ 0768	ݸ 0778
9	ݙ 0759	ݩ 0769	ݹ 0779
A	ݚ 075A	ݪ 076A	ݺ 077A
B	ݛ 075B	ݫ 076B	ݻ 077B
C	ݜ 075C	ݬ 076C	ݼ 077C
D	ݝ 075D	ݭ 076D	ݽ 077D
E	ݞ 075E	ݮ 076E	ݾ 077E
F	ݟ 075F	ݯ 076F	ݿ 077F

08A0 **Arabic Extended-A** **08FF**



FB50 **Arabic Presentation Forms-A** **FC3F**

	FB5	FB6	FB7	FB8	FB9	FBA	FBB	FBC	FBD	FBE	FBF	FC0	FC1	FC2	FC3
0	ٱ FB50	ٺ FB60	ٿ FB70	ڀ FB80	ڙ FB90	ٹ FBA0	ٻ FBB0	ٰ FBC0		ە FBE0	ۉ FBF0	ئج FC00	ئح FC10	ئم FC20	ئى FC30
1	ٱ FB51	ٺ FB61	ٿ FB71	ڀ FB81	ڙ FB91	ٹ FBA1	ٻ FBB1	ٰ FBC1		ە FBE1	ۉ FBF1	ئح FC01	ئج FC11	ئم FC21	ئى FC31
2	ٻ FB52	ټ FB62	ٿ FB72	ڀ FB82	ڙ FB92	ٹ FBA2	ٻ FBB2			ۆ FBE2	ۉ FBF2	ئم FC02	ئم FC12	ضج FC22	ئي FC32
3	ٻ FB53	ټ FB63	ٿ FB73	ڀ FB83	ڙ FB93	ٹ FBA3	ٻ FBB3		ڤ FBD3	ۆ FBE3	ۉ FBF3	ئى FC03	ئى FC13	ضح FC23	قح FC33
4	ٻ FB54	ټ FB64	ٿ FB74	ڀ FB84	ڙ FB94	ٹ FBA4	ٻ FBB4		ڤ FBD4	ې FBE4	ۋ FBF4	ئي FC04	ثي FC14	ضخ FC24	قم FC34
5	ٻ FB55	ټ FB65	ٿ FB75	ڀ FB85	ڙ FB95	ٹ FBA5	ٻ FBB5		ڭ FBD5	ې FBE5	ۋ FBF5	بج FC05	جح FC15	ضم FC25	قى FC35
6	پ FB56	ٽ FB66	ٿ FB76	ڀ FB86	ڙ FB96	ٹ FBA6	ٻ FBB6		ڭ FBD6	ۑ FBE6	ۋ FBF6	بح FC06	جم FC16	طح FC26	قي FC36
7	پ FB57	ٽ FB67	ٿ FB77	ڀ FB87	ڙ FB97	ٹ FBA7	ٻ FBB7		ۇ FBD7	ۑ FBE7	ۇ FBF7	بخ FC07	حج FC17	طم FC27	كا FC37
8	پ FB58	ٽ FB68	ٿ FB78	ڀ FB88	ڙ FB98	ٹ FBA8	ٻ FBB8		ۇ FBD8	ـ FBE8	ۈ FBF8	بم FC08	حم FC18	ظم FC28	كج FC38
9	پ FB59	ٽ FB69	ٿ FB79	ڀ FB89	ڙ FB99	ٹ FBA9	ٻ FBB9		ۈ FBD9	ـ FBE9	ۈ FBF9	بى FC09	خج FC19	عج FC29	كح FC39
A	ٻ FB5A	ٽ FB6A	ٿ FB7A	ڀ FB8A	ڙ FB9A	ٹ FBAA	ٻ FBBA		ۈ FBDA	ـ FBEA	ۈ FBFA	بي FC0A	خح FC1A	عم FC2A	كخ FC3A
B	ٻ FB5B	ٽ FB6B	ٿ FB7B	ڀ FB8B	ڙ FB9B	ٹ FBAB	ٻ FBBB		ۋ FBDB	ئا FBEB	ئد FBFB	تج FC0B	خم FC1B	غج FC2B	كل FC3B
C	ٻ FB5C	ٽ FB6C	ٿ FB7C	ڀ FB8C	ڙ FB9C	ٹ FBAC	ٻ FBBC		ۋ FBDC	ئە FBEC	ى FBFC	تح FC0C	سج FC1C	غم FC2C	كم FC3C
D	ٻ FB5D	ٽ FB6D	ٿ FB7D	ڀ FB8D	ڙ FB9D	ٹ FBAD	ٻ FBBD		ۋ FBDD	ئە FBED	ى FBFD	تخ FC0D	سح FC1D	فج FC2D	كى FC3D
E	ٺ FB5E	ٽ FB6E	ٿ FB7E	ڀ FB8E	ڙ FB9E	ٹ FBAE	ٻ FBBE		ۅ FBDE	ئو FBEE	ي FBFE	تم FC0E	سخ FC1E	فح FC2E	كى FC3E
F	ٺ FB5F	ٽ FB6F	ٿ FB7F	ڀ FB8F	ڙ FB9F	ٹ FBAF	ٻ FBBF		ۅ FBDF	ئو FBEF	ي FBFF	تى FC0F	سم FC1F	فخ FC2F	لج FC3F

FC40 **Arabic Presentation Forms-A** **FD1F**

	FC4	FC5	FC6	FC7	FC8	FC9	FCA	FCB	FCC	FCD	FCE	FCF	FD0	FD1
0	لح FC40	ني FC50	◌ّ◌َ NBSP FC60	تر FC70	كا FC80	ى' FC90	بهـ FCA0	سمـ FCB0	فخـ FCC0	مخـ FCD0	ئهـ FCE0	يمـ FCF0	حي FD00	ضر FD10
1	لخ FC41	هج FC51	◌ّ◌ُ NBSP FC61	تز FC71	كل FC81	ير FC91	تجـ FCA1	صحـ FCB1	فمـ FCC1	ممـ FCD1	بمـ FCE1	يهـ FCF1	جى FD01	طى FD11
2	لم FC42	هم FC52	◌ّ◌ِ NBSP FC62	تم FC72	كم FC82	ين FC92	تحـ FCA2	صخـ FCB2	قحـ FCC2	نجـ FCD2	بهـ FCE2	ـَّـ FCF2	جي FD02	طي FD12
3	لى FC43	هى FC53	◌ّ◌ٕ NBSP FC63	تن FC73	كى FC83	يم FC93	تخـ FCA3	صمـ FCB3	قمـ FCC3	نحـ FCD3	تمـ FCE3	ـُّـ FCF3	خى FD03	عى FD13
4	لي FC44	هي FC54	ئر FC64	تى FC74	كي FC84	ين FC94	تمـ FCA4	ضجـ FCB4	كجـ FCC4	نخـ FCD4	تهـ FCE4	ـِّـ FCF4	خي FD04	عي FD14
5	مج FC45	يج FC55	ئز FC65	تي FC75	لم FC85	يى FC95	تهـ FCA5	ضحـ FCB5	كحـ FCC5	نمـ FCD5	ثمـ FCE5	طى FCF5	صى FD05	غى FD15
6	مح FC46	يح FC56	ئم FC66	ثر FC76	لى FC86	يي FC96	ثمـ FCA6	ضخـ FCB6	كخـ FCC6	نهـ FCD6	ثهـ FCE6	طي FCF6	صي FD06	غي FD16
7	مخ FC47	يخ FC57	ئن FC67	ثز FC77	لي FC87	ئج FC97	جج FCA7	ضمـ FCB7	كلـ FCC7	هجـ FCD7	سمـ FCE7	عى FCF7	ضى FD07	سى FD17
8	مم FC48	يم FC58	ئى FC68	ثم FC78	ما FC88	ئح FC98	جمـ FCA8	طحـ FCB8	كمـ FCC8	همـ FCD8	سهـ FCE8	عي FCF8	ضي FD08	سي FD18
9	مى FC49	يى FC59	ئي FC69	ثن FC79	مم FC89	ئخ FC99	حجـ FCA9	ظمـ FCB9	لجـ FCC9	هٰـ FCD9	شمـ FCE9	غى FCF9	شج FD09	شى FD19
A	مي FC4A	يي FC5A	بر FC6A	ثى FC7A	نر FC8A	ئم FC9A	حمـ FCAA	عجـ FCBA	لحـ FCCA	يجـ FCDA	شهـ FCEA	غي FCFA	شح FD0A	شي FD1A
B	نج FC4B	ذٰ FC5B	بز FC6B	ثي FC7B	نز FC8B	ئه FC9B	خجـ FCAB	عمـ FCBB	لخـ FCCB	يحـ FCDB	كلـ FCEB	سى FCFB	شخ FD0B	حى FD1B
C	نح FC4C	رٰ FC5C	بم FC6C	فى FC7C	نم FC8C	بج FC9C	خمـ FCAC	غجـ FCBC	لمـ FCCC	يخـ FCDC	كمـ FCEC	سي FCFC	شم FD0C	حي FD1C
D	نخ FC4D	ىٰ FC5D	بن FC6D	في FC7D	نن FC8D	بح FC9D	سجـ FCAD	غمـ FCBD	لهـ FCCD	يمـ FCDD	لمـ FCED	شى FCFD	شر FD0D	جى FD1D
E	نم FC4E	◌ّ◌ٖ NBSP FC5E	بى FC6E	قى FC7E	نى FC8E	بخ FC9E	سحـ FCAE	فجـ FCBE	مجـ FCCE	يهـ FCDE	نمـ FCEE	شي FCFE	سر FD0E	جي FD1E
F	نى FC4F	◌ّ◌ٗ NBSP FC5F	بي FC6F	قي FC7F	ني FC8F	بم FC9F	سخـ FCAF	فحـ FCBF	محـ FCCF	ئمـ FCDF	نهـ FCEF	حى FCFF	صر FD0F	خى FD1F

FD20 **Arabic Presentation Forms-A** **FDFF**

	FD2	FD3	FD4	FD5	FD6	FD7	FD8	FD9	FDA	FDB	FDC	FDD	FDE	FDF
0	ﳢ FD20	ﳲ FD30		ﵐ FD50	ﵠ FD60	ﵰ FD70	ﶀ FD80		﶐ FDA0	ﶠ FDB0	ﶰ FDC0			ﷀ FDF0
1	ﳣ FD21	ﳳ FD31		ﵑ FD51	ﵡ FD61	ﵱ FD71	ﶁ FD81		﶑ FDA1	ﶡ FDB1	ﶱ FDC1			ﷁ FDF1
2	ﳤ FD22	ﳴ FD32		ﵒ FD52	ﵢ FD62	ﵲ FD72	ﶂ FD82	ﶒ FD92	ﶒ FDA2	ﶢ FDB2	ﶲ FDC2			ﷲ FDF2
3	ﳥ FD23	ﳵ FD33		ﵓ FD53	ﵣ FD63	ﵳ FD73	ﶃ FD83	ﶓ FD93	ﶓ FDA3	ﶣ FDB3	ﶳ FDC3			ﷳ FDF3
4	ﳦ FD24	ﳶ FD34		ﵔ FD54	ﵤ FD64	ﵴ FD74	ﶄ FD84	ﶔ FD94	ﶔ FDA4	ﶤ FDB4	ﶴ FDC4			ﷴ FDF4
5	ﳧ FD25	ﳷ FD35		ﵕ FD55	ﵥ FD65	ﵵ FD75	ﶅ FD85	ﶕ FD95	ﶕ FDA5	ﶥ FDB5	ﶵ FDC5			ﷵ FDF5
6	ﳨ FD26	ﳸ FD36		ﵖ FD56	ﵦ FD66	ﵶ FD76	ﶆ FD86	ﶖ FD96	ﶖ FDA6	ﶦ FDB6	ﶶ FDC6			ﷶ FDF6
7	ﳩ FD27	ﳹ FD37		ﵗ FD57	ﵧ FD67	ﵷ FD77	ﶇ FD87	ﶗ FD97	ﶗ FDA7	ﶧ FDB7	ﶷ FDC7			ﷷ FDF7
8	ﳪ FD28	ﳺ FD38		ﵘ FD58	ﵨ FD68	ﵸ FD78	ﶈ FD88	ﶘ FD98	ﶘ FDA8	ﶨ FDB8		ﷸ FDD8	ﷸ FDE8	ﷸ FDF8
9	ﳫ FD29	ﳻ FD39		ﵙ FD59	ﵩ FD69	ﵹ FD79	ﶉ FD89	ﶙ FD99	ﶙ FDA9	ﶩ FDB9		ﷹ FDD9	ﷹ FDE9	ﷹ FDF9
A	ﳬ FD2A	ﳼ FD3A		ﵚ FD5A	ﵪ FD6A	ﵺ FD7A	ﶊ FD8A	ﶚ FD9A	ﶚ FDAA	ﶪ FDBA		ﷺ FDDA	ﷺ FDEA	ﷺ FDFA
B	ﳭ FD2B	ﳽ FD3B		ﵛ FD5B	ﵫ FD6B	ﵻ FD7B	ﶋ FD8B	ﶛ FD9B	ﶛ FDAB	ﶫ FDBB		ﷻ FDDB	ﷻ FDEB	ﷻ FDFB
C	ﳮ FD2C	ﳾ FD3C		ﵜ FD5C	ﵬ FD6C	ﵼ FD7C	ﶌ FD8C	ﶜ FD9C	ﶜ FDAC	ﶬ FDBC		﷼ FDDC	﷼ FDEC	﷼ FDFC
D	ﳯ FD2D	ﳿ FD3D		ﵝ FD5D	ﵭ FD6D	ﵽ FD7D	ﶍ FD8D	ﶝ FD9D	ﶝ FDAD	ﶭ FDBD		﷽ FDDD	﷽ FDED	﷽ FDFD
E	ﳰ FD2E	ﴀ FD3E		ﵞ FD5E	ﵮ FD6E	ﵾ FD7E	ﶎ FD8E	ﶞ FD9E	ﶞ FDAE	ﶮ FDBE		﷾ FDDE	﷾ FDEE	
F	ﳱ FD2F	ﴁ FD3F		ﵟ FD5F	ﵯ FD6F	ﵿ FD7F	ﶏ FD8F	ﶟ FD9F	ﶟ FDAF	ﶯ FDBF		﷿ FDDF	﷿ FDEF	

FE70 **Arabic Presentation Forms-B** **FEFF**

	FE7	FE8	FE9	FEA	FEB	FEC	FED	FEE	FEF
0	◌ً FE70	ء FE80	ب FE90	ج FEA0	ز FEB0	ض FEC0	غ FED0	ـل FEE0	ى FEF0
1	ـً FE71	آ FE81	بـ FE91	ح FEA1	س FEB1	ط FEC1	ف FED1	ـم FEE1	ي FEF1
2	◌ٌ FE72	آ FE82	بـ FE92	ح FEA2	س FEB2	ط FEC2	ف FED2	ـم FEE2	ي FEF2
3	، FE73	أ FE83	ة FE93	حـ FEA3	سـ FEB3	ط FEC3	فـ FED3	مـ FEE3	يـ FEF3
4	◌ٍ FE74	أ FE84	ة FE94	حـ FEA4	سـ FEB4	ط FEC4	غـ FED4	مـ FEE4	يـ FEF4	
5		ؤ FE85	ت FE95	خ FEA5	ش FEB5	ظ FEC5	ق FED5	ن FEE5	لآ FEF5
6	◌َ FE76	ؤ FE86	تـ FE96	خـ FEA6	شـ FEB6	ظـ FEC6	قـ FED6	نـ FEE6	للآ FEF6
7	ـَ FE77	إ FE87	تـ FE97	خـ FEA7	شـ FEB7	ظـ FEC7	قـ FED7	نـ FEE7	لآ FEF7
8	◌ُ FE78	إ FE88	تـ FE98	خـ FEA8	شـ FEB8	ظـ FEC8	قـ FED8	نـ FEE8	للآ FEF8
9	ـُ FE79	ئ FE89	ث FE99	د FEA9	ص FEB9	ع FEC9	ك FED9	ه FEE9	لآ FEF9
A	◌ِ FE7A	ئ FE8A	ثـ FE9A	دـ FEAA	صـ FEBA	عـ FECA	كـ FEDA	هـ FEEA	للآ FEFA
B	ـِ FE7B	ئ FE8B	ثـ FE9B	ذ FEAB	صـ FEBB	عـ FECB	كـ FEDB	هـ FEEB	لا FEFB
C	◌ّ FE7C	ئ FE8C	ثـ FE9C	ذ FEAC	صـ FEBC	عـ FECC	كـ FEDC	هـ FEEC	لا FEFC
D	ـّ FE7D	ا FE8D	ج FE9D	ر FEAD	ض FEBD	غ FECD	ل FEDD	و FEED	
E	◌ْ FE7E	ا FE8E	جـ FE9E	رـ FEAE	ضـ FEBE	غـ FECE	لـ FEDE	وـ FEEE	
F	ـْ FE7F	ب FE8F	جـ FE9F	ز FEAF	ضـ FEBF	غـ FECF	لـ FEDF	ى FEEF	FEFF

6.4 A Large Glyph Sample

It's useful to look at a few letters up close to examine some of the subtleties in the design. In particular, the tapering towards extremities is the opposite of traditional Arabic design, in keeping with the Knuthian spirit, but it still has a culturally authentic feel.