ALI-BABA AND THE 4.0 UNICODE CHARACTERS
New input and output concepts under Unicode

Thomas Milo
The Netherlands
tmilo@decotype.com

ABSTRACT

New input and output concepts under Unicode: Literary, Classical and Qur'ānic Arabic in Unicode involve more characters than the Arabic keyboard can accommodate. To enter sophisticated Arabic, Basis Technology and DecoType jointly developed an IME (input method editor): ALI (Arabic, Latin Input). When used at full power, ALI overstretches conventional font technology. Only DecoType's Arabic Calligraphic Engine ACE, dubbed BABA for the occasion, handles elaborate ALI output. Therefore, the ALI-BABA pair points the way toward technology for dealing with Arabic text –from dialect transcriptions to the most complicated Qur'ānic quotation, covering all aspects of both abstract text level and rendering it to the highest possible standards of computer generated typography.

RÉSUMÉ

Des nouveaux concepts d'entrée et de sortie sous Unicode. L'arabe litérraire, classique et coranique requiert plus de caractères que le clavier de saisie arabe ne peut fournir. Pour saisir de l'arabe «sophistiqué», les socieétés Basis Technology et Decotype ont développé en commun un EME (Éditeur de méthode d'entrée) du nom de ALI (Arabic, Latin Input). Utilisé à tout son potentiel, ALI dépasse la technologie conventionnelle de fonte. Seule la technologie d'arabe calligraphique de DecoType (appelée ici BABA pour l'occasion), est en mesure de gérer efficacement la sortie de ALI. C'est pour cette raison que le couple ALI-BABA montre la voie vers une nouvelle technologie de composition arabe – des transcriptions dialectales jusqu'aux citations coraniques les plus complexes, en couvrant tous les aspects du niveau textuel abstrait et en affichant ce texte aux plus hauts standards de typographie informatique.

Editor's note: This article is rather different in style from the rest of the proceedings, as the author is not a TEX user, and prepared it with the tools he normally uses. Due to the nature of the material and pressing publication deadlines, we felt it best to print it this way, rather than take the considerable additional time that would have been necessary to typeset it in the customary fashion.

INTRODUCTION

The extent of Arabic script

The scope of Arabic or Islamic script is very broad indeed. By definition it covers the Islamic civilization, both in its historical development and in its geographical expansion.

For literary and scholarly publishing all orthographical units or graphemes are relevant, as are all stages of the orthography of the Arabic language through the ages as well as all contemporary and historical orthographies of all other languages that use Arabic script or once did so.

Computing in Arabic

In its early stages in the '70s of the last century, Arabic computing was totally focused on contemporary governmental and short term business needs. As input a small subset of Arabic graphemes sufficed; output quality, let alone any typographical quality, was of no relevance. Against that background, Arabic computing took off with the typewriter metaphor: A very limited keyboard (*input*) with a matching set of glyphs (*output*). Precision of spelling (input) and typographical accuracy (output) in Arabic computing are primitive when compared with the quality of Arabic typesetting that it is now necessarily superseding.

Figure 1: The typewriter metaphor – simplified input.

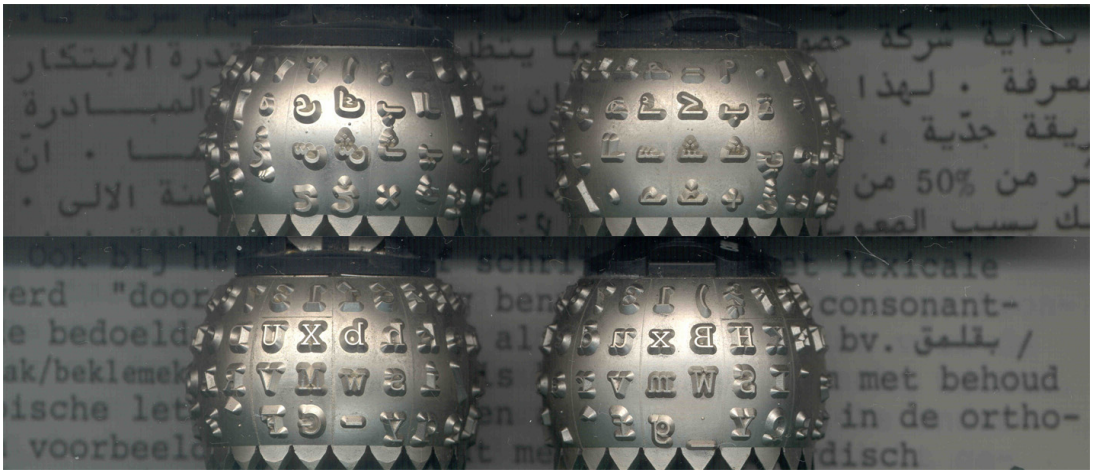


Figure 2: The typewriter metaphor – simplified output:
only two glyphs per key, using the shift mechanism that was designed to handle upper case in Latin script.

In the 19th and early 20th century, the typesetter of Arabic used no keyboard.
For input and output, the typographer had in front of him four or more cases with metal glyphs capable of representing even the most heavily annotated orthography and designed to match the quality of well-written manuscripts.

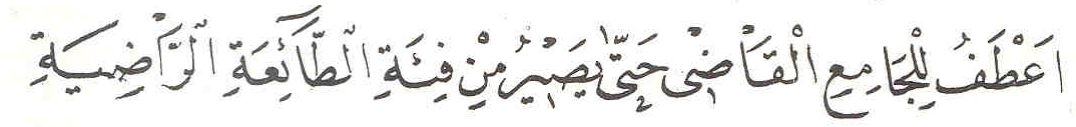


Figure 3: Fragment of nasḫ *writing taken from an Ottoman manuscript*
Computer typesetting to date can not yet register all graphemes used nor can it match its graphic quality.

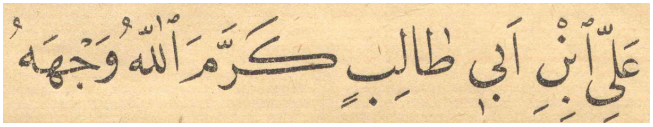


Figure 4: Fragment of nasḫ *typesetting taken from an Ottoman book.*
Note the use of superscript and subscript alifs in this particular spelling variant.

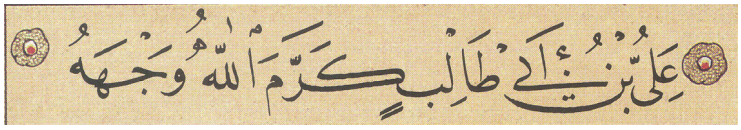


Figure 5: Nasḫ *writing by Mustafa Izzet Efendi, the 19th century Ottoman top calligrapher whose hand was the model for the typeface in figure 4.*

These glyphs represented anything from ligatures, allographs, parts of letters and superscript/subscript annotation marks (vowels, Qur'ānic punctuation).

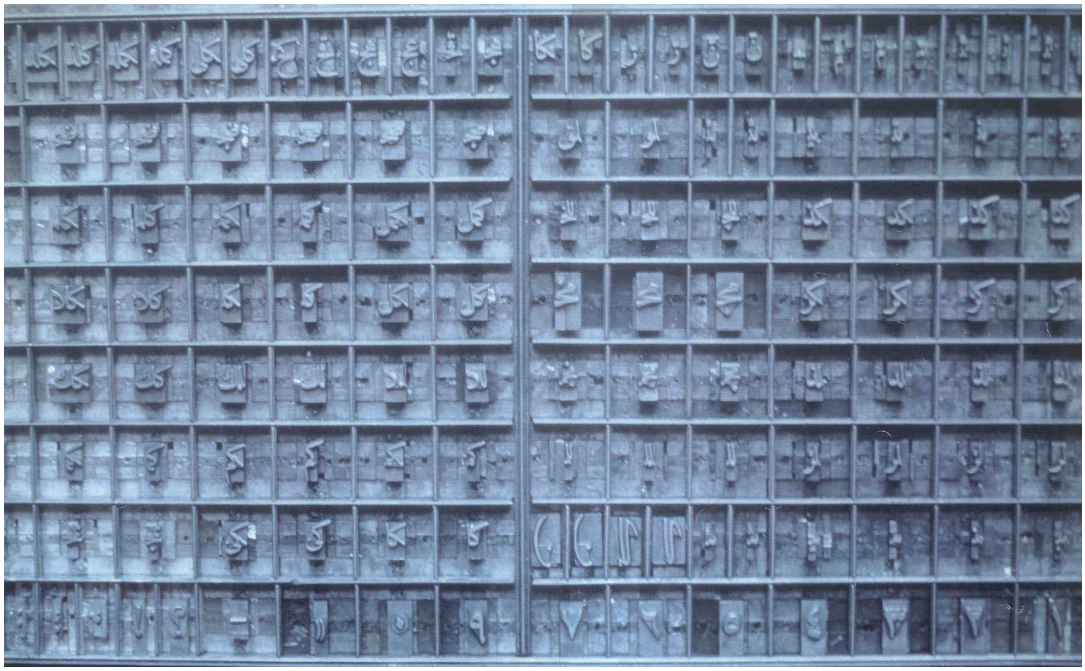


Figure 6: Detail of a case of movable types used in Egypt to typeset the Royal Qur'ān (1920's). The total glyph set amounted to about 500 pieces capturing some three dozen graphemes in their contextual variation. There was no mechanism to select the correct contextual variant (allograph): it all depended on the skill of the typesetter and his familiarity with calligraphic practice.

Today, in an age where technology allows man to walk on the Moon, for Arabic publishing the transition from mechanical typesetting to computer typography had the paradoxical effect of imposing severe limitations and causing graphic quality to plummet: input no longer handles sophisticated spelling and output no longer follows the classic esthetics of *nasḫ* script accurately.

INPUT — MORE GRAPHEMES

The present repertoire of Arabic characters in Unicode is about to reach the point that Ottoman typography reached in the 19th century: it can handle Literary Arabic, Classical Arabic and Qur'ānic Arabic — theoretically. But making practical use of this latest, richer set is no sinecure: Unicode has outgrown the keyboard based on the typewriter metaphor.

In order to make the larger repertoire of Arabic in Unicode accessible, Basis Technology and DecoType jointly developed the ALI — Arabic, Latin Input. It is an IME (input method editor) based on proven CJK (Chinese-Japanese-Korean) IME technology. ALI allows you to enter complex, fully vocalized Arabic orthography with a Latin keyboard.

This IME uses a straightforward Latin transcription. The method of transcription provides *linguistic* information and should not be confused with transliteration, which provides *graphic* information. Since the spelling principles of Arabic script differ considerably from that of any Latin orthography, transliteration tends to be unpronounceable. Transcription aims to be pronounceable.

To clarify the difference between transliteration and transcription we need to compare the Latin representation of the phrase shown in figure 3 (the computing limitations under scrutiny here, actually impose a slightly altered spelling[1]):

أَعْطَفُ لِلْجَامِعِ ٱلْقَاضِي حَتَّى يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

Transliterated in ASCII:

OaEoTafu lilojaAmiEi {loqaADiy Hat~aY yaSiyru mino fi}api {lT~aA}iEapi {lr~aADiyapi

This is a meticulous but unpronounceable representation of the script.[2]

Transcribed in ASCII:

a`Tafu li l-jaami`i l-qaaDii Hattae yaSiiru min fi'äti T-Taa'i`äti r-raaDiyäti

This is an accurate representation of speech with little direct correspondence with the Arabic scripted original.[3] A comparison with the transliteration sample shows that it actually requires less Latin characters.

Our design aim was to stay within the ASCII-set (the key codes common to all computer keyboards). When transcription is typed – including all vowels using Latin characters — the correct Arabic spellings are computed simultaneously and rendered in Arabic script. In this

[1] As the reader proceeds, he will notice a number of serious Arabic font errors in the typeset text. The "guinea pig" font used for the examples is the DecoType Naskh typeface designed to Windows 95 specifications and unfortunately not capable of handling vowels and ligatures according to the rules of *nasḫ* typography: vowels do not follow to skeleton text, print on top of each other and misplaced lengthening bars (*taṭwīl — kešīde*) cause the font to behave erroneously. Moreover, the font will display default squares instead of the latest Unicode additions. These design limitations are representative for the state of the art and in this treatise they are used to illustrate the argumentation. In the second part, this font is contrasted with the DecoType Authentic Naskh typeface, designed to ACE (Arabic Calligraphic Engine) specifications: this form of computerized Arabic script not subject to traditional constraints.

II: American Standard Code for Information Interchange. The tranliteration shown is designed for computer use by Tim Buckwalter, to enable data interchange in this smallest computing character set available using even left and right parentheses to represent Arabic letters. (http://www.qamus.org/transliteration.htm).

[3] ALI converts the ASCII sequence *a+* (for *tā' marbūṭä*) automatically to the mnemonic *ä.*

process orthographic complexities and spelling rules are automatically resolved. E.g., the sound *hamz* is typed as a single apostrophe /'/ regardless the contextually varying spelling in the Arabic script:

ijraa'u-n	إِجْرَاءٌ
ijraa'uhu	إِجْرَاؤُهُ
ijraa'ahu	إِجْرَاءَهُ
ijraa'ihi	إِجْرَائِهِ
ijraa'aatu-n	إِجْرَاءَاتٌ
ijraa'iyyu-n	إِجْرَائِيٌّ

The transcription method used for ALI is designed to enable automatic conversion to and from Arabic orthography: a full roundtrip. As all inspected existing Latin-based transcription methods[4] are ambivalent in certain aspects (for instance, final -a and -ah cannot be reliably converted), disambiguation needed to be introduced in such cases. For instance, the final **/a/** in the transcription of **/layla/** requires additional precision.

conventional	ALI	modern Arabic
layla	**layla**	لَيْلَ
layla	**laylä**	لَيْلَة
layla	**laylae**	لَيْلَى
layla	**layla-n**	لَيْلًا (لَيْلاً)

Since the use of annotated vowels in Arabic is optional, vowel generation can be turned off partly or completely. ALI offers a configuration screen to reduce the amount of annotation. This enables the user to generate different degrees of orthographic precision in Arabic from one and the same Latin transcription:

Fully vocalized

أَعْطَفُ لِلْجَامِعِ ٱلْقَاضِي حَتَّى يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

[4] http://ee.www.ee/transliteration/pdf/Arabic.pdf.

Waṣlä removed

أَعْطَفُ لِلْجَامِعِ الْقَاضِي حَتَّى يَصِيرُ مِنْ فِئَةِ الطَّائِعَةِ الرَّاضِيَةِ

Vowels removed

أعطف للجامع القاضي حتّى يصير من فئة الطّائعة الرّاضية

Shadda and *alif-hamzä* removed

اعطف للجامع القاضي حتى يصير من فئة الطائعة الراضية

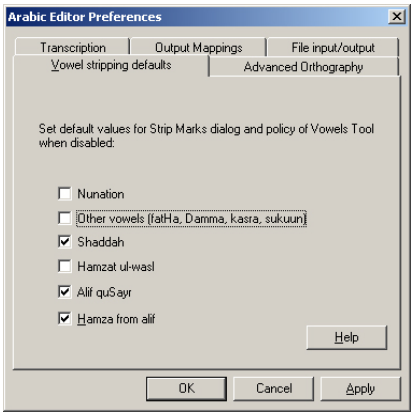


Figure 7: ALI-*based Arabic Editor's vowel stripping controls allow the Arabic representation to become simpler*[5].

Being an Input Method Editor, ALI hinges on Arabic transcription instead of Arabic script. This facilitates conversion from one regional typographical norm to another. A classical example is the presence or absence of dots under the Arabic letter *yā'* in final position, or the stripping of dots on *tā' marbūṭä* and vice versa without loss of information. A much less known fact is that fully annotated Arabic can have variations in types of vowel marks. For this, too, ALI offers a configuration screen to manage the amount of sophistication that is required. Again, from one and the same Latin transcription:

Standard computer orthography

أَعْطَفُ لِلْجَامِعِ ٱلْقَاضِي حَتَّى يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

Conventional typography has no dots on final *yā'*

أَعْطَفُ لِلْجَامِعِ ٱلْقَاضِى حَتَّى يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

Older spellings often don't use *hamzä* with vocalized *alif*

اَعْطَفُ لِلْجَامِعِ ٱلْقَاضِى حَتَّى يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

[5] This setting can also be developed for ALI to generate so-called archigraphemic Arabic representation, i.e., the archaic ambivalent skeleton text without dots and vowels.

Superscript *alif* to differentiate *alif maqṣūrä* from *yā'*

اَعْطَفُ لِلْجَامِعِ ٱلْقَاضِى حَتَّىٰ يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

Extra *sukūn* for long vowels

اَعْطَفُ لِلْجَامِعِ ٱلْقَاضِى حَتَّىٰ يَصِيرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

Subscript "dagger" *alif* for long *kasrä* (not supported by many fonts)

اَعْطَفُ لِلْجَامِعِ ٱلْقَاضْ□ى حَتَّىٰ يَصِ□يرُ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

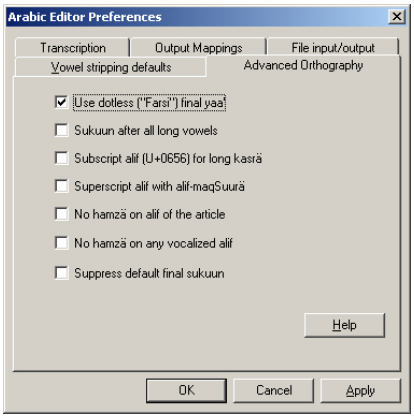


Figure 8: ALI*-based Arabic Editor's vowel elaboration controls allow the Arabic spelling to be more complex.*

With a simple expansion ALI can also provide transcriptions in any academic convention like those adopted by the US Library of Congress, British Library, E.J. Brill's Encyclopaedia of Islam, die Deutsche Morgenländische Gesellschaft, etc. etc. For instance:

reversible ASCII transcription, connecting vowels, linked words:
a`Tafu li l-jaami`i l-qaaDii Hattae yaSiiru min fi'äti T-Taa'i`äti r-raaDiyäti

reversible typographic transcription, connecting vowels, linked words:
aʿṭafu li l-jāmiʿi l-qāḍī ḥattae yaṣīru min fiʾäti ṭ-ṭāʾiʿäti r-rāḍiyäti

reversible typographic transcription librarian's style, separate words:
aʿṭafu li-l-jāmiʿ ʾl-qāḍī ḥattae yaṣīru min fiʾäti ʾl-ṭāʾiʿäh ʾl-rāḍiyäh

irreversible standard transcription, librarian's style, separate words:
aʿṭafu li-l-jāmiʿ ʾl-qāḍī ḥattā yaṣīru min fiʾäti ʾl-ṭāʾiʿah ʾl-rāḍiyah

On the other hand, ALI can compute the transcription of text in Arabic script. This is a very powerful feature that can be used to create variant spellings in Arabic to suppress misprints caused by legacy font incompatibilities. Depending on the value of the setting Rendering Compatibility Conventions, variant spellings can be generated – again! – from one and the same transcription:

Correct order (*lām-fatḥatān-alif*)

layla-n لَيْلًا

Tweaked order (*lām-alif- fatḥatān*) to avoid breaking up of *lām-alif*

layla-n لَيْلاً

In publishing, this feature of ALI can help text editors to stabilize the order in which Arabic characters are encoded. Moreover, in the case of fully vocalized Arabic, ALI makes it possible to find Arabic letters that lack a vowel marker:

(أَعْطَفُ لِلْجَامِعِ ٱلْقَاضِي حَتَّى يَصيرُ مِن فِئَة ٱلطَّائِعَة ٱلرَّاضِيَة)

Transcription of fully annotated text

a`Tafu li l-jaami`i l-qaaDii Hattae yaSiiru min fi'äti T-Taa'i`äti r-raaDiyäti

Defective annotation revealed by underscore

a`Tafu li l-jaami`i l-qA_Dii Hattae yaS_y_ru min_ fi'ä T-Taa'i`äti r-raaDiyäti

NOTE - Transcribed Arabic contains more textual information than Arabic orthography. By introducing a graphically transparent morpheme separator for Arabic Unicode text (comparable to U+00AD Soft Hyphen) it would be possible to retain transcription data that would otherwise be lost in Arabic spelling, e.g.: to prevent **/li l-jaami`i/** to revert to **/liljaami`i/**.

A totally different aspect of ALI is that it can be used to apply elongations, known by the Arabic term *taṭwīl* or the Persian *kešīde*, without having to fall back on Arabic Keyboard mode. And, since such esthetic elongation has no status in transcription, ALI automatically strips a word of *taṭwīl* when one brings up the Latin transcription.[6].

OUTPUT — MORE ALLOGRAPHS

When used for fully vocalized Classical Arabic, the Unicode text produced by ALI exceeds the capabilities of computer fonts. In fact, present font technology bars the production of this level of Arabic text.

اَعْطَفُ لِلْجَامِعِ ٱلْقَاضْ□ىْ حَتَّىْ يَص□ْيْرُ مِنْ فِئَة ٱلطَّائْعَة ٱلرَّاْضِيَة

This is where we pull a special card out of our sleeve: ACE, the Arabic Calligraphic Engine, for the occasion dubbed BABA to match ALI. This technology has the advantage that it was conceived with Classical Arabic in mind from the beginning:

1. ACE covers all Arabic Unicodes required for the most sophisticated forms of Qur'ānic, Classical and Literary Arabic;
2. ACE can handle any combination or layer of annotation marks;
3. ACE resolves clashes between annotation marks dynamically;

[6] Ideally, aesthetic elongation doesn't belong in text code, as it is covered by the font rendering mechanism when creating block justification in text layout. A practical example of this can be seen in Adobe InDesign ME with DecoType fonts (there is is called "Naskh justification").

اَعْطَفُ لِلْجَامِعِ الْقَاضِي حَتَّىٰ يَصِيرُ مِنْ فِئَةِ الطَّائِعَةِ الرَّاضِيَةِ

Where ALI produces variant sequences to suppress legacy font quirks, ACE technology intercepts these variant sequences to stabilize results (DT copyrighted). But this technology can do more: traditional writing methods are sometimes inseparable from conventional spellings, therefore:

4. ACE optimizes Unicode strings according to the conventions required for the selected typeface. For instance, modern spellings often place the Fathatan on top of the trailing *alif*, instead of before it. For a font inspired by the *muḥaqqaq* style this would be inappropriate indeed. In such cases ACE is capable of suppressing any Rendering Compatibility Conventions (legacy font tweaks) generated by ALI:

Figure 8: Qur'ân fragment(21:69: "…be cool and safe for…" in muḥaqqaq calligraphy by Aḥmad Ibn as-Suhrawardi[7].

ALI transcription

kuunii barda-n wa salaama-n `alae

ALI output in full orthography with *fatḥatān* (twin strokes) before *alif*:

كُوْنِى بَرْدًا وَسَلَامًا عَلَىْ

ALI output in reduced ("font kludge") orthography with *fatḥatān* on *alif*:

كُوْنِى بَرْدا وَسَلَامًا عَلَىْ

ACE output with typeface-specific correction of the *fatḥatān* in the lower line:

كُوْنِى بَرْدًا وَسَلَامًا عَلَىْ

كُوْنِى بَرْدًا وَسَلَامًا عَلَىْ

As mentioned in the lasts paragraph of the section about input, Unicode has a code point called *taṭwīl* (U+0640). Here we encounter the typewriter metaphor at its narrowest: the typewriter had a lengthening bar and now the computer industry has a lengthening bar, too. This crowbar-like glyph becomes a spanner in the works of contextual shaping when used in final position or following discontinuous letters (e.g., *rā'*/*zain*).

Taṭwīl is in fact an Arabic translation of the Persian-Ottoman calligraphic notion of *kešīde* "stretched", "lengthened" (and actually called *madd* by Arab calligraphers). When applied on a connection between letters it creates an elegant curve; when applied on certain final letters it gives them an esthetically pleasing swash.

ACE technology aims to treat *taṭwīl* as a *kešīde* in the true sense of the word. Therefore it uses a pair of novel technologies to make the daily use of the "lengthening bar" fool-proof:

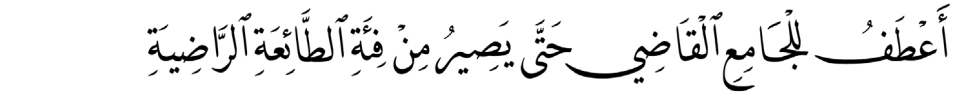
[7] See: www.islamicart.com/main/calligraphy/late.html.

5. ACE secures the calligraphically correct execution of a *taṭwīl* or *kešīde*. *Trashideh*© technology (DT copyrighted) prevents, in a typeface-specific way, the execution of *taṭwīl* codes in calligraphically illegal positions, e.g., in initial position or, in the case of *ruqʿä* script, in all positions:

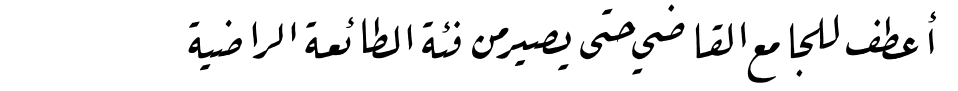
Normal font rendering any Unicode 0640 as a lengthening bar

أَـــعْطَفُـــ لِلْجَامِعِـــ ٱلْقَاضِيـــ حَتَّى يَصِيرُـــ مِنْ فِئَةِ ٱلطَّائِعَةِ ٱلرَّاضِيَةِ

ACE-generated *nasḫ* output with typeface-specific *kešīde* in legal positions only



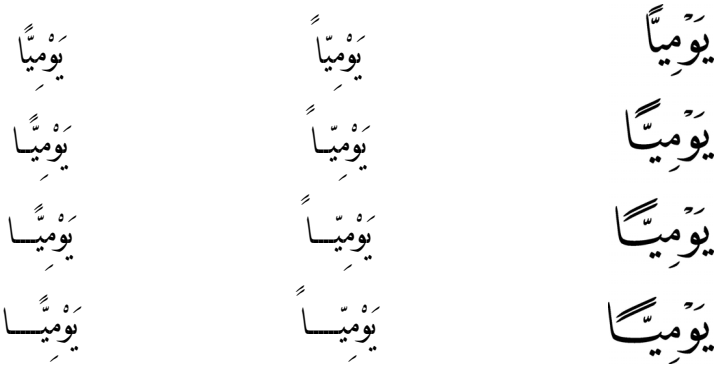
ACE-generated *ruqʿä* output with typeface-specific omission of *kešīde*[8]



Finally, computer-generated elongation is defeated by the failure to shape vowels accordingly. Here is a solution that gets more mileage out of the elongation bar:

6. ACE technology provides for proportional annotation marks that automatically match the underlying stretched forms (DT copyrighted).

classic orthography and "font kludge" spelling straightened out by ACE



CONCLUSION

The ALI-BABA pair offers a new perspective for technology dealing with Arabic text — from the simplest office letter to the most complicated Qur'ānic quotation, covering all aspects of the abstract text level, and rendering it to the highest possible standards of computer generated typography[9].

Thomas Milo
Amsterdam, June 2003

[8] *Ruqʿä* script allows *kešīde* in final position, but the in the DecoType *Ruqʿä* used for this illustration this is not yet implemented.

[9] The transcription concept described in this article is implemented in the Basis technology Arabic Editor. An evaluation copy can be downloaded from this URL: http://www.basistech.com/arabic-editor/.