# Proposal to encode productive Arabic-script modifier marks

**Date:** May 16, 2003  
**Author:** Jonathan Kew, SIL International  
**Address:** Horsleys Green, High Wycombe, Bucks HP14 3XL, England, England  
**Tel:** +44 (1494) 682306  
**Email:** jonathan_kew@sil.org  

**Kamal Mansour,** Agfa Monotype  
**Mark Davis,** IBM  
**Address:** 425 Sherman Ave, Palo Alto, CA 94306, USA  
**Tel:** +1 (650) 321-4102  
**Email:** kamal.mansour@agfamonotype.com, mark.davis@jtcsv.com  

## A. Administrative

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Title</td>
</tr>
<tr>
<td>2.</td>
<td>Requester's name</td>
</tr>
<tr>
<td>3.</td>
<td>Requester type</td>
</tr>
<tr>
<td>4.</td>
<td>Submission date</td>
</tr>
<tr>
<td>5.</td>
<td>Requester's reference</td>
</tr>
<tr>
<td>6a.</td>
<td>Completion</td>
</tr>
<tr>
<td>6b.</td>
<td>More information to be provided?</td>
</tr>
</tbody>
</table>

## B. Technical — General

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1a.</td>
<td>New script? Name?</td>
</tr>
<tr>
<td>1b.</td>
<td>Addition of characters to existing block? Name?</td>
</tr>
<tr>
<td>2.</td>
<td>Number of characters in proposal</td>
</tr>
<tr>
<td>3.</td>
<td>Proposed category</td>
</tr>
<tr>
<td>4.</td>
<td>Proposed level of implementation and rationale</td>
</tr>
<tr>
<td>5a.</td>
<td>Character names included in proposal?</td>
</tr>
<tr>
<td>5b.</td>
<td>Character names in accordance with guidelines?</td>
</tr>
<tr>
<td>5c.</td>
<td>Character shapes reviewable?</td>
</tr>
<tr>
<td>6a.</td>
<td>Who will provide computerized font?</td>
</tr>
<tr>
<td>6b.</td>
<td>Font currently available?</td>
</tr>
<tr>
<td>6c.</td>
<td>Font format?</td>
</tr>
<tr>
<td>7a.</td>
<td>Are references (to other character sets, dictionaries, descriptive texts, etc.) provided?</td>
</tr>
<tr>
<td>7b.</td>
<td>Are published examples (such as samples from newspapers, magazines, or other sources) of use of proposed characters attached?</td>
</tr>
<tr>
<td>8.</td>
<td>Does the proposal address other aspects of character data processing?</td>
</tr>
</tbody>
</table>
## C. Technical — Justification

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Has this proposal for addition of character(s) been submitted before?</td>
<td>No (but see L2/02-021 for a related proposal).</td>
<td></td>
</tr>
<tr>
<td>2a. Has contact been made to members of the user community?</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>2b. With whom?</td>
<td>Academics working with Arabic script; language communities in South Asia and North Africa.</td>
<td></td>
</tr>
<tr>
<td>3. Information on the user community for the proposed characters is included?</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>4. The context of use for the proposed characters</td>
<td>Current use in publications in languages of North Africa, South Asia, and Near &amp; Middle East; experimental orthographies for previously unwritten minority languages; scholarly and pedagogical use.</td>
<td></td>
</tr>
<tr>
<td>5. Are the proposed characters in current use by the user community?</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>6a. Must the proposed characters be entirely in the BMP?</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>6b. Rationale?</td>
<td>Contemporary characters in current use</td>
<td></td>
</tr>
<tr>
<td>7. Should the proposed characters be kept together in a contiguous range?</td>
<td>Preferably (for convenience of users and implementers), but not essential.</td>
<td></td>
</tr>
<tr>
<td>8a. Can any of the proposed characters be considered a presentation form of an existing character or character sequence?</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>8b. Rationale for inclusion?</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>9a. Can any of the proposed characters be considered to be similar (in appearance or function) to an existing character?</td>
<td>Several proposed characters (#1, #2, #8-#10, #19-#22) appear similar to combining marks in the U+03xx block. Character #7 looks similar to U+0615. Character #10 looks similar to U+0652. Character #23 looks similar to U+06BA.</td>
<td></td>
</tr>
<tr>
<td>9b. Rationale for inclusion?</td>
<td>Characteristic appearance of dots used in Arabic script differs from generic combining marks; different combining classes needed from U+03xx marks. U+0615 is a higher-level (phrasal) mark, and U+0652 is a diacritic that operates at the level of the vowel marks; neither is a component used in creating new letters. #23 differs from U+06BA in joining behavior.</td>
<td></td>
</tr>
<tr>
<td>10. Does the proposal include the use of combining characters and/or use of composite sequences?</td>
<td>Yes, all but one of the proposed characters are combining</td>
<td>Combining marks</td>
</tr>
<tr>
<td>11. Does the proposal contain characters with any special properties?</td>
<td>Combining marks</td>
<td></td>
</tr>
</tbody>
</table>

## D. SC2/WG2 Administrative

To be completed by SC2/WG2

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Relevant SC2/WG2 document numbers</td>
<td>GenArabProp.tex</td>
<td>16-May-2003</td>
</tr>
<tr>
<td>2. Status (list of meeting number and corresponding action or disposition)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Additional contact to user communities, liaison organizations, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Assigned category and assigned priority/time frame</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other comments</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
I. Background

The Universal Character Set currently encodes Arabic letters as indivisible units. However, the structure of the script is better understood as a small set of “skeletal” letterforms, to which modifier marks (primarily patterns of dots, but other marks are also used) are added to differentiate additional sounds (or letters) as needed to write a particular language. The creation of specific base letter + modifier combinations is a living, productive process, still in use today as Arabic script is adopted for writing additional languages.

To more adequately model and support the productive nature of modifier mark usage in Arabic, it is proposed that a set of combining modifier characters be added to the UCS Arabic repertoire. In addition to allowing users to encode any combination of skeletal Arabic letterform plus modifiers, supporting the real-world evolutionary use of the script in transcribing new languages, this model also enables scholars to encode historical documents where modifiers may have been added over time, and allows the modifiers to be encoded individually where required (e.g., in pedagogical materials). In these, it may be necessary to apply markup separately to base characters and modifiers; this cannot generally be done using only precomposed letters.

While supporting a set of combining Arabic modifiers does require some effort on the part of implementers, especially of rendering systems and fonts, it is not substantially different in nature or complexity from that already required to properly support the Arabic vowel diacritics, Koranic marks, etc. Moreover, both standardization bodies and implementers will benefit from the fact that the Arabic block can be stabilized more quickly in this way than by encoding each new combination of base letter + modifiers in its own right, as and when it is proposed and documented as an individual new character.

II. Proposal

1. Repertoire

The following list of characters are proposed as additions to the UCS; all are combining marks, with the exception of #23.

For each modifier mark, one or more examples are given of letters where the mark has been used as a modifier on an undotted skeletal Arabic letterform. Where possible, examples of such “precomposed” letters from the current UCS repertoire are included. In addition, for most of the marks there are examples showing how they have been used to create new letters for minority languages whose orthographies are not yet well standardized or documented.

The letters and languages mentioned here are merely representative of the widespread practice of “extending” the Arabic script in this manner; additional examples can be found in the references listed at the end of this document.

<table>
<thead>
<tr>
<th>Number</th>
<th>Character</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>⠤ ⠧</td>
<td>ARABIC MODIFIER SINGLE DOT ABOVE</td>
<td>⠠Ứ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>2.</td>
<td>⠤ ⠥</td>
<td>ARABIC MODIFIER TWO DOTS HORIZONTALLY ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>3.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER TWO DOTS VERTICALLY ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>4.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER THREE DOTS UPWARDS ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>5.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER THREE DOTS DOWNWARDS ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>6.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER FOUR DOTS ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>7.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER TAH ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>8.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER SMALL V ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>9.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER INVERTED V ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>10.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER RING ABOVE</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
<tr>
<td>11.</td>
<td>⠤ ⠤</td>
<td>ARABIC MODIFIER SINGLE DOT BELOW</td>
<td>⠠祇 ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧ ⠧</td>
</tr>
</tbody>
</table>
At the time of writing, there are fewer than 30 unallocated positions in the Arabic block at U+06xx; with 23 new characters in the current proposal, and keeping in mind that there are known to be a number of additional vowels and other diacritics still to be encoded, it seems clear that the 06xx block is going to overflow. It is suggested, therefore, that ARABIC LETTER DOTLESS NOON be encoded in the existing Arabic block (perhaps at 065F), but that a separate extension block be allocated for the modifier marks. For the convenience of implementers, it would be helpful for this to be close to the 06xx block. One possibility would be to move one of the tentatively-allocated scripts in the Roadmap to space higher in the BMP, to allow creation of an Arabic Modifiers block.

2. Properties

All the proposed characters except for #23 ARABIC LETTER DOTLESS NOON are non-spacing combining marks, with General Category Mn and bidirectional type NSM. They have no numeric values, are not mirrored, and have no decompositions.

It has been suggested that they should all have combining class 0, and therefore would not participate in canonical reordering of combining characters. This is attractive in its simplicity, and allows users to encode any sequence knowing that no process will modify the ordering of the modifier marks. However, it has the disadvantage that where modifiers are used both above and below a single base letter, there would be multiple visually indistinguishable sequences and canonical ordering would not address this situation.

It is therefore proposed that these new Arabic combining marks be given new, low-value combining classes. (They should not use the existing classes in the 200s, as this would put them further from the base letter than vowel points during canonical ordering, greatly complicating rendering and other processes.) The assignments proposed are:

- Class 1 (already used for combining overlays): #22 ARABIC MODIFIER BAR THROUGH LETTER.
- Class 2: Arabic modifier marks below the base character (#11-#21).
- Class 3: Arabic modifier marks above the base character (#1-#10).

The one base character proposed here, ARABIC LETTER DOTLESS NOON, has General Category Lo, combining class 0, bidirectional type AL, and no other special properties. Its Arabic joining class is NOON. It differs from U+06BA ARABIC LETTER NOON GHUNNA in that U+06BA in initial and medial forms acquires a single dot above; the proposed ARABIC LETTER DOTLESS NOON, being intended as a base for the Arabic modifier marks, should not exhibit this behavior.

3. Usage notes and normalization issues

As a general rule, a skeletal (“dotless”) Arabic letter may be modified by the addition of one or more of these modifier marks, to form a new extended Arabic-script consonant (or occasionally a vowel, in some
languages). Marks may be used both above and below the same letter (as illustrated by some of the precomposed letters already in Unicode); it is also permissible to stack multiple marks either above or below, as shown in a few of the examples above.

Implementers (in particular, font developers) should note that certain combinations of base letter + modifier may require special positioning. This can be seen in a number of the extended Arabic characters already encoded, such as \texttt{U+0696}, where the position of the ARABIC MODIFIER SINGLE DOT ABOVE is different from that seen on \texttt{U+0632}. Another example is the use of the ARABIC MODIFIER RING BELOW, typically written attached to the base letter as in Pashto \texttt{U+067C}. The ARABIC MODIFIER RING ABOVE is seen attached to the base letter in \texttt{U+06C4}, but is written separately in Punjabi \texttt{U+0685} (in one of several competing orthographies).

With the encoding of a set of Arabic modifier marks, it will be possible to represent many Arabic-script letters in two ways: either using the existing precomposed letters available in the U+06xx block, or using dotless letters plus modifier marks. As the existing precomposed letters have no decompositions (and stability policies prohibit adding them), the two forms will not be canonically equivalent. This means that there will be multiple “spellings” possible for Arabic script text, and Unicode normalization processes will not treat them as equivalent. It is strongly recommended that the existing precomposed letters should be used wherever possible, with sequences involving the new modifier marks being used only to encode letters that are not otherwise available.

An additional twist to the question of precomposed versus decomposed representations is that it would be possible to add modifier marks to precomposed letters that already incorporate modifiers of their own, such as adding the mark \#4 \texttt{ARABIC MODIFIER THREE DOTS UPWARDS ABOVE} to \texttt{U+062C ARABIC LETTER JEEM} to form the Wolof letter \texttt{U+062C ARABIC LETTER JEEM} (Alternatively, this same letter could be created with mark \#11 \texttt{ARABIC MODIFIER SINGLE DOT BELOW} applied to \texttt{U+0685}.) In the absence of canonical equivalence between the precomposed and decomposed forms, these would represent different “spellings” of the same extended Arabic letter. However, it is strongly recommended that users avoid such mixtures of precomposed and decomposed representation, and always use a fully-decomposed sequence for any letter that is not directly available as a precomposed form.

A data file, \texttt{DiscArabCombSeq.txt}, is provided (see Appendix A) that lists all the skeletal Arabic letters to which the modifier marks may be added. Use of these marks with other base characters, although not illegal, is considered non-standard practice and is unlikely to be well supported by fonts and rendering systems. The file also lists the sequences of skeletal letters and modifier marks that are strongly discouraged because they should be visually indistinguishable from existing Arabic letters. Arabic text processing systems may wish to offer users the option to convert between these sequences and the corresponding precomposed letters, or otherwise treat them as equivalent for certain purposes.

Implementers can encourage usage in accordance with these recommendations through the keyboard layouts or other input methods that are provided for languages with letters that must be composed using sequences. Implementations could even provide options to detect and warn users if these marks are applied to base characters other than the expected skeletal Arabic letters, and if sequences listed as “discouraged” in \texttt{DiscArabCombSeq.txt} are found. (For example, rendering systems could have the ability to display such combinations in visibly distinct ways, such as with marks serialized instead of stacked.)

4. Collation

For the purposes of the Unicode Collation Algorithm, it is proposed that the modifier marks be treated as Level 1 ignorables in the Default Unicode Collation Element Table, and given Level 2 weights. This means that all letters based on the same skeletal form will sort together in the default ordering at the primary level.

However, it is assumed that for any language where a new extended Arabic letter, encoded using a sequence of base + modifier marks, is used as part of the alphabet, the collation sequence would be tailored to sort the specific sequences used into their proper alphabetical positions.

A data file, \texttt{ArabicModifierKeys.txt}, is provided (see Appendix B) that lists proposed weights for the modifier marks, designed for use as an extension to version 3.1.1 of the standard \texttt{allkeys.txt} file. Note that ARABIC LETTER DOTLESS NOON should be inserted with a Level 1 weight immediately following U+0646 ARABIC LETTER NOON, affecting all following Level 1 weights. This will be only one of a number of changes with global effect required for a complete UCA update; the data given here illustrates how the new modifier marks would be integrated but does not represent a complete update to the Default Unicode Collation Element Table.
III. Chart of proposed characters

In addition to the characters shown here, it is proposed to insert \textless{} ARABIC LETTER DOTLESS NOON at U+065F (or another available codepoint) in the existing Arabic block. No specific codepoints are shown for the modifier marks, but it is suggested that they be kept near the existing Arabic block for the convenience of implementers.
IV. Names list

Addition to Arabic block

065F  ﻤ ARABIC LETTER DOTLESS NOON

Arabic modifier marks above

xx00  ﻠ ARABIC MODIFIER SINGLE DOT ABOVE
xx01  ﻠ ARABIC MODIFIER TWO DOTS HORIZONTALLY ABOVE
xx02  ﻠ ARABIC MODIFIER TWO DOTS VERTICALLY ABOVE
xx03  ﻠ ARABIC MODIFIER THREE DOTS UPWARDS ABOVE
xx04  ﻠ ARABIC MODIFIER THREE DOTS DOWNWARDS ABOVE
xx05  <reserved>
xx06  ﻠ ARABIC MODIFIER FOUR DOTS ABOVE
xx07  ﻠ ARABIC MODIFIER TAH ABOVE
xx08  ﻠ ARABIC MODIFIER SMALL V ABOVE
xx09  ﻠ ARABIC MODIFIER INVERTED V ABOVE
xx0A  ﻠ ARABIC MODIFIER RING ABOVE
xx0B  <reserved>
xx0C  <reserved>
xx0D  <reserved>
xx0E  <reserved>
xx0F  <reserved>

Arabic modifier marks below

xx10  ﻠ ARABIC MODIFIER SINGLE DOT BELOW
xx11  ﻠ ARABIC MODIFIER TWO DOTS HORIZONTALLY BELOW
xx12  ﻠ ARABIC MODIFIER TWO DOTS VERTICALLY BELOW
xx13  ﻠ ARABIC MODIFIER THREE DOTS UPWARDS BELOW
xx14  ﻠ ARABIC MODIFIER THREE DOTS DOWNWARDS BELOW
xx15  ﻠ ARABIC MODIFIER THREE DOTS HORIZONTALLY BELOW
xx16  ﻠ ARABIC MODIFIER FOUR DOTS BELOW
xx17  ﻠ ARABIC MODIFIER TAH BELOW
xx18  ﻠ ARABIC MODIFIER SMALL V BELOW
xx19  ﻠ ARABIC MODIFIER INVERTED V BELOW
xx1A  ﻠ ARABIC MODIFIER RING BELOW
xx1B  <reserved>
xx1C  <reserved>
xx1D  <reserved>
xx1E  <reserved>
xx1F  <reserved>

Arabic modifier marks through

xx20  ﻠ ARABIC MODIFIER BAR THROUGH LETTER
V. References


Appendix A: Expected base characters & discouraged sequences

The machine-readable data file DiscArabCombSeq.txt provides a list of all the skeletal Arabic letters that are normally expected to be used as base characters for the modifier marks. Then it also lists the specific sequences beginning with these base characters that are discouraged in normal use, as they represent letters that are already encoded as individual Unicode characters.

The draft data file listed here uses abbreviated names to refer to proposed characters for which Unicode codepoints are not yet available.

Listing of data file DiscArabCombSeq.txt
# This file documents "expected" and "discouraged" usage for the Arabic-script modifier marks.
# There are three sections to the file.
# First, there is a list of the characters classified as "Arabic-script modifier marks".
# Second, a list of all the characters that are considered normal bases for the modifier marks; using any of the modifier marks on base characters not in this list is considered non-standard.
# and is strongly discouraged unless there is a clear need and no alternative code sequence.
# Third, there is a list of sequences involving an "expected" base plus one or more modifier marks that are strongly discouraged, because they would be visually identical to pre-existing Arabic script characters. Such sequences should never be interchanged.
# On each line, there are two data fields separated by semicolon. Field 1 gives one or more Unicode scalar values; field 2 contains a code indicating which of the three types of code or sequence is represented by field 1.
# Field 2 format:
# M: arabic modifier marks
# B: normal base character for application of arabic modifier marks
# X: sequences that are strongly discouraged
# Characters considered "expected" base characters for the Arabic-script modifier marks:

# Characters classified as Arabic-script modifier marks:
<one dot above> : M # ARABIC MODIFIER SINGLE DOT ABOVE
<two dots horiz above> : M # ARABIC MODIFIER TWO DOTS HORIZONTALLY ABOVE
<two dots vert above> : M # ARABIC MODIFIER TWO DOTS VERTICALLY ABOVE
<three dots up above> : M # ARABIC MODIFIER THREE DOTS UPWARDS ABOVE
<three dots down above> : M # ARABIC MODIFIER THREE DOTS DOWNWARDS ABOVE
<three dots horiz below> : M # ARABIC MODIFIER THREE DOTS HORIZONTALLY BELOW
<four dots above> : M # ARABIC MODIFIER FOUR DOTS ABOVE
<table above> : M # ARABIC MODIFIER TAH ABOVE
<small v above> : M # ARABIC MODIFIER SMALL V ABOVE
<inverted v above> : M # ARABIC MODIFIER INVERTED V ABOVE
<br>above> : M # ARABIC MODIFIER RING ABOVE
<one dot below> : M # ARABIC MODIFIER SINGLE DOT BELOW
<two dots horiz below> : M # ARABIC MODIFIER TWO DOTS HORIZONTALLY BELOW
<two dots vert below> : M # ARABIC MODIFIER TWO DOTS VERTICALLY BELOW
<three dots up below> : M # ARABIC MODIFIER THREE DOTS UPWARDS BELOW
<three dots down below> : M # ARABIC MODIFIER THREE DOTS DOWNWARDS BELOW
<three dots horiz below> : M # ARABIC MODIFIER THREE DOTS HORIZONTALLY BELOW
<four dots below> : M # ARABIC MODIFIER FOUR DOTS BELOW
<table below> : M # ARABIC MODIFIER TAH BELOW
<small v below> : M # ARABIC MODIFIER SMALL V BELOW
<inverted v below> : M # ARABIC MODIFIER INVERTED V BELOW
<br>below> : M # ARABIC MODIFIER RING BELOW
<br>through> : M # ARABIC MODIFIER BAR THROUGH LETTER
# Characters considered "expected" base characters for the Arabic-script modifier marks:
0621 : B # ARABIC LETTER HAMZA
0627 : B # ARABIC LETTER ALEF
062D : B # ARABIC LETTER MAH
062F : B # ARABIC LETTER DAL
0631 : B # ARABIC LETTER REH
0633 : B # ARABIC LETTER SEEN
0635 : B # ARABIC LETTER SAD
0637 : B # ARABIC LETTER TAH
0639 : B # ARABIC LETTER AIN
0643 : B # ARABIC LETTER KAF
0644 : B # ARABIC LETTER LAM
0645 : B # ARABIC LETTER MEEM
Sequences that are discouraged in normal use, being indistinguishable from preexisting letters:

- \text{dotless noon} \# Questionable because of use of \text{064A} in canonical decomposition of 0626.
- \text{0649} \# ARABIC LETTER YEH \# ***

# ARABIC LETTER DOTLESS NOON

- \text{0649} \# ARABIC LETTER YEH WITH TAIL
- \text{06D2} \# ARABIC LETTER YEH BARREE
- \text{06D5} \# ARABIC LETTER AE
<three dots down below> <three dots up above> : X # 069C # ARABIC LETTER SEEN WITH THREE DOTS BELOW AND THREE DOTS ABOVE
<three dots up above> <three dots down below> : X # 069C # ARABIC LETTER SEEN WITH THREE DOTS BELOW AND THREE DOTS ABOVE
<two dots horiz below> : X # 069D # ARABIC LETTER SAD WITH TWO DOTS BELOW
<four dots above> : X # 069E # ARABIC LETTER TAH WITH FOUR DOTS ABOVE
<three dots up above> : X # 069F # ARABIC LETTER TAH WITH THREE DOTS ABOVE
<three dots up above> : X # 06A0 # ARABIC LETTER AIN WITH THREE DOTS ABOVE
<one dot below> : X # 06A1 # ARABIC LETTER FEH WITH DOT MOVED BELOW
<one dot below> : X # 06A2 # ARABIC LETTER FEH WITH DOT MOVED BELOW
<one dot below> : X # 06A3 # ARABIC LETTER FEH WITH DOT BELOW
<one dot below> : X # 06A4 # ARABIC LETTER VEH
<one dot below> : X # 06A5 # ARABIC LETTER FEH WITH THREE DOTS BELOW
<four dots above> : X # 06A6 # ARABIC LETTER PEHEH
<one dot above> : X # 06A7 # ARABIC LETTER QAF WITH DOT ABOVE
<three dots up above> : X # 06A8 # ARABIC LETTER QAF WITH THREE DOTS ABOVE
<three dots up above> : X # 06A9 # ARABIC LETTER KAF WITH THREE DOTS ABOVE
<one dot above> : X # 06AA # ARABIC LETTER KAF WITH DOT ABOVE
<one dot above> : X # 06AB # ARABIC LETTER LAM WITH SMALL V
<one dot above> : X # 06AC # ARABIC LETTER KAF WITH DOT ABOVE
<one dot above> : X # 06AD # ARABIC LETTER NG
<three dots up above> : X # 06AE # ARABIC LETTER KAF WITH THREE DOTS BELOW
<ring below> : X # 06B0 # ARABIC LETTER GAF WITH RING
<two dots horiz above> : X # 06B1 # ARABIC LETTER NGOEH
<two dots horiz below> : X # 06B2 # ARABIC LETTER GAF WITH TWO DOTS BELOW
<two dots vert below> : X # 06B3 # ARABIC LETTER GUEH
<three dots up above> : X # 06B4 # ARABIC LETTER GAF WITH THREE DOTS ABOVE
<small v above> : X # 06B5 # ARABIC LETTER LAM WITH SMALL V
<one dot above> : X # 06B6 # ARABIC LETTER LAM WITH DOT ABOVE
<three dots up above> : X # 06B7 # ARABIC LETTER LAM WITH THREE DOTS BELOW
<dotless noon> <one dot below> <one dot above> : X # 06B9 # ARABIC LETTER NOON WITH DOT BELOW
<dotless noon> <one dot below> <one dot above> : X # 06BA # ARABIC LETTER NOON WITH DOT BELOW
<dotless noon> <dotless noon> <one dot below> <one dot above> : X # 06BB # ARABIC LETTER RNOWN
<dotless noon> <ring below> <one dot above> <one dot below> : X # 06BC # ARABIC LETTER NOON WITH RING
<dotless noon> <one dot above> <one dot below> <two dots horiz above> : X # 06BD # ARABIC LETTER NOON WITH THREE DOTS BELOW

Excluded because of non-standard linking behavior: dots go below in initial and medial forms.
<two dots horiz below> : X # 06BF # ARABIC LETTER TCHEH WITH DOT ABOVE
<three dots down below> : X # 06C0 # ARABIC LETTER TAH WITH THREE DOTS ABOVE
<two dots horiz above> : X # 06C2 # ARABIC LETTER MARBUTA GOAL
<two dots horiz below> : X # 06C3 # ARABIC LETTER TEH MARBUTA GOAL
<ring above> : X # 06C4 # ARABIC LETTER WAH WITH RING
<bar through> : X # 06C5 # ARABIC LETTER KIRGHIZ OE
<small v above> : X # 06C6 # ARABIC LETTER OE
<inverted v above> : X # 06C9 # ARABIC LETTER KIRGHIZ YU
<two dots horiz above> : X # 06CA # ARABIC LETTER WAH WITH TWO DOTS ABOVE
<four dots above> : X # 06CB # ARABIC LETTER VE
<small v above> : X # 06CE # ARABIC LETTER YEH WITH SMALL V
<one dot below> : X # 06CF # ARABIC LETTER YEH WITH SMALL V
<two dots vert below> : X # 06D0 # ARABIC LETTER E
<three dots down below> : X # 06D1 # ARABIC LETTER YEH WITH THREE DOTS BELOW
<inverted v above> : X # 06EE # ARABIC LETTER DAL WITH INVERTED V
<one dot below> : X # 06EF # ARABIC LETTER REH WITH INVERTED V
<three dots up above> : X # 06FA # ARABIC LETTER SHEEN WITH DOT BELOW
<one dot below> : X # 06FB # ARABIC LETTER DAD WITH DOT BELOW
<one dot below> : X # 06FC # ARABIC LETTER GHAIN WITH DOT BELOW
<one dot below> : X # 06FD # ARABIC LETTER GHAIN WITH DOT BELOW
<inverted v above> : X # 06FF # ARABIC LETTER HEH WITH INVERTED V
Appendix B: Default collation keys

Suggested collation key values for the proposed Arabic modifier marks are given in the file ArabicModifierKeys.txt. The values used here are based on those found in version 3.1.1 of the standard allkeys.txt file.

The draft data file listed here uses abbreviated names to refer to proposed characters for which Unicode codepoints are not yet available.

Listing of data file ArabicModifierKeys.txt

# Suggested default key weights for Arabic modifier marks, based on version 3.1.1 of allkeys.txt
#
# Individual characters; 4th field to be Unicode codepoint
#

<single dot above> : [.0000.0200.0002.XXXX] # ARABIC MODIFIER SINGLE DOT ABOVE
<two dots above> : [.0000.0201.0002.XXXX] # ARABIC MODIFIER TWO DOTS HORIZONTALLY ABOVE
<three dots above> : [.0000.0202.0002.XXXX] # ARABIC MODIFIER THREE DOTS UPWARDS ABOVE
<four dots above> : [.0000.0204.0002.XXXX] # ARABIC MODIFIER THREE DOTS DOWNWARDS ABOVE
<two dots horiz above> : [.0000.0201.0002.XXXX] # ARABIC MODIFIER TWO DOTS HORIZONTALLY ABOVE
<two dots vert above> : [.0000.0202.0002.XXXX] # ARABIC MODIFIER TWO DOTS VERTICALLY ABOVE
<three dots up above> : [.0000.0203.0002.XXXX] # ARABIC MODIFIER THREE DOTS UPWARDS ABOVE
<three dots down above> : [.0000.0204.0002.XXXX] # ARABIC MODIFIER THREE DOTS DOWNWARDS ABOVE
<two dots horiz below> : [.0000.0221.0002.XXXX] # ARABIC MODIFIER TWO DOTS HORIZONTALLY BELOW
<two dots vert below> : [.0000.0222.0002.XXXX] # ARABIC MODIFIER TWO DOTS VERTICALLY BELOW
<three dots up below> : [.0000.0223.0002.XXXX] # ARABIC MODIFIER THREE DOTS UPWARDS BELOW
<three dots down below> : [.0000.0224.0002.XXXX] # ARABIC MODIFIER THREE DOTS DOWNWARDS BELOW
<three dots horiz below> : [.0000.0226.0002.XXXX] # ARABIC MODIFIER THREE DOTS HORIZONTALLY BELOW
<four dots below> : [.0000.0227.0002.XXXX] # ARABIC MODIFIER FOUR DOTS BELOW
<single dot below> : [.0000.0220.0002.XXXX] # ARABIC MODIFIER SINGLE DOT BELOW
<inverted v above> : [.0000.0212.0002.XXXX] # ARABIC MODIFIER INVERTED V ABOVE
<inverted v below> : [.0000.0225.0002.XXXX] # ARABIC MODIFIER INVERTED V BELOW
<ring above> : [.0000.0213.0002.XXXX] # ARABIC MODIFIER RING ABOVE
<ring below> : [.0000.0233.0002.XXXX] # ARABIC MODIFIER RING BELOW
<bar through> : [.0000.0240.0002.XXXX] # ARABIC MODIFIER BAR THROUGH
<bar through> : [.0000.0241.0002.XXXX] # ARABIC MODIFIER BAR THROUGH
<bar through> : [.0000.0242.0002.XXXX] # ARABIC MODIFIER BAR THROUGH
<dotless noon> : [.0F34.0020.0002.XXXX] # ARABIC LETTER DOTLESS NOON

# DOTLESS NOON goes in the main sequence right after ARABIC LETTER NOON.
# All Level 1 weights from 0F34 (ARABIC LETTER NOON GHUNNA) upwards will then be increased by 1.
#