

# Orthography issues and Unicode

In Arabic script

2017 ©SIL International

Now that we've looked at the sociolinguistic issues for orthography decisions we should look at Unicode related issues.

Open Tavultesoft Character Identifier

Open <http://graphicore.github.io/charset-inspector/>

Open folder CharSetInspector

## Character / Glyph model

U+0628 ب ب ب

U+0648 و و

U+06CC ي ي ي

U+0644 + U+0627 ل ا ل

2017 ©SIL International

You are all familiar with Arabic script and so I expect you to understand Unicode a bit better than the average person. However, I want to start with a few basics to make sure we are on the same page.

Let's start with the letter beh U+0628. We have the isolate form and we have initial, medial and final forms. Do they get stored in the computer the same way? Yes, they are all stored as U+0628. The font, along with the operating system or application figures out which form to use based on where it appears in the word.

We call U+0628 a character. Any of the 4 forms are the character. However, there are 4 glyphs involved. The isolate is one glyph, the initial another, and so on.

Looking at it simplistically, if there is a space before U+0628 and a space after U+0628 it will give the isolate form. If there is a space before U+0628 and another character after U+0628, then it will give the initial form. If there are characters on either side of U+0628 it will give the medial form and if there is a character before and a space after it will give the final form.

Of course, there are other factors, such as combining marks, such as vowels, shadda, sukun, etc which must be factored in. Punctuation instead of spaces comes into play.

Is everyone with me so far?

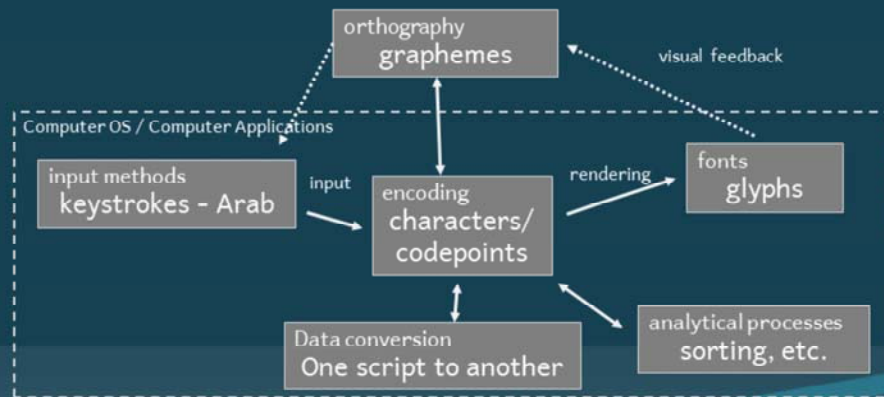
The beh is called a dual-joining character. Some characters in Arabic are right-joining only, such as the waw and the reh. You know the waw only has an isolate and final form. It doesn't have initial or medial forms. Both forms are still stored in the computer as U+0648.

These are the easier ones. They follow a pattern. Let's look at a character called the farsi yeh. The initial and medial forms have dots, but the isolate and final forms do not. Some people have chosen to use different characters (codepoints) for the different glyphs. However, these should still be stored in the computer as U+06CC.

Lastly we look at a ligature. It is stored in the computer as two codepoints, but it is displayed as one glyph.

Why does this matter?

## Chart overview and how they interact



2017 ©SIL International

Separately we looked at orthography issues from a sociolinguistic perspective. That must be settled first, but it all must work within a computing system and that revolves around the encoding. We must have the encoding settled before we can create an input method. We must have the encoding to develop fonts to support and we must have the encoding to run analytical processes on our texts.

This week we will be looking at each part of this chart from different perspectives.

## Coming up with a new orthography

- You have choices:
  - Same sounds as other languages – use same character
  - New/different sounds:
    - Use an existing character in Unicode
    - Create a new character
    - Using other characters to create new characters
    - Use different codepoints depending on the position in the word
    - Using hidden characters

2017 ©SIL International

Here is an overview of options for coming up with a new orthography.

I realize some of you already have a settled orthography and some of you are still deciding. I believe this is helpful information to understand.

## Use an existing character in Unicode

- Look at what exists in Unicode
  - Charts: <http://www.unicode.org/charts/>

- (Never use the Presentation Forms block for encoded characters. These are there for historical purposes.)

پ	ڈ	پس	ک	ن	و
067B	068B	069B	06AB	06BB	06CB
ت	ڈ	پش	ک	ن	ی
067C	068C	069C	06AC	06BC	06CC
ت	ی	ص	ک	ن	ی
067D	068D	069D	06AD	06BD	06CD
پ	ڈ	ض	پ	ه	ی
067E	068E	069E	06AE	06BE	06CE

2017 ©SIL International

There are many, many different characters in Unicode and there should be options for you to select a character that already exists. The Unicode charts are a good starting place.

## Create a new character

- Try not to do this!
  - However, if the characters have been in use for several years it might be worth proposing to Unicode

2017 ©SIL International

Some people will want to create new characters to represent the sounds in their language. While this might sound good there are long term negative ramifications which we will address next. However, we should make it clear that we **can** propose new characters to Unicode if there is evidence the character is widely accepted. It just takes a very long time for approval and then implementation in fonts, operating systems and applications.

## Create a new character (cont.)

- Examples

- U+08A7  ARABIC LETTER MEEM WITH THREE

### DOTS ABOVE

- Chadian request
- Proposed 10-Aug-2010
- Published in Unicode 6.1 (31-Jan-2012)
- Added to Scheherazade (18-Sep-2012)
- Supported in Win 8.1 (08-Apr-2014)
- Supported in Office 2016

In 2010 30+ characters were proposed for adding to Unicode. Let's take a look at two of the characters.

ARABIC LETTER MEEM WITH THREE DOTS ABOVE was a fairly straightforward request. It had official support from the Chadian government and it was clearly a new character and not a composition of two existing characters. You can see it took about 4 years from the first proposal before it was supported in the Windows operating system.



## Create a new character (cont.)

- Examples

- U+08A1 ﺀ ARABIC LETTER BEH WITH HAMZA ABOVE

- Cameroon request
    - Initially proposed 10-Aug-2010
    - Reproposed: 25-Oct-2010
    - Added to Scheherazade (18-Sep-2012)
    - Published in Unicode 7.0 (16-Jun-2014)
    - Supported in Win 10 (29-Jul-2015)
    - Supported inOffice 2016
  - Adobe InDesign still does not support these characters. The newest version supports through Unicode 6.0
  - <http://software.sil.org/arabicfonts/support/application-support/>

2017 ©SIL International

ARABIC LETTER BEH WITH HAMZA ABOVE was proposed at the same time as the other character. It was quite controversial because there is already a beh in Unicode and a combining hamza. It represents an implosive b. You can see that it took approximately 5 years from the initial proposal until it was supported in Windows.

Neither of these characters are yet supported in InDesign!

We've compiled a fairly comprehensive list of commonly used applications on Windows and OSX. We've tried to document the level of Unicode support for each version of Unicode.

## Create a new character (cont.)

- Why was ARABIC LETTER BEH WITH HAMZA ABOVE required?
  - Eg, why not just use a BEH + combining hamza?
  - A single consonant
    - requires possibility of vowels, shadda, sukun above
  - No association with the HAMZA; it has merely borrowed its graphic form

Why was ARABIC LETTER BEH WITH HAMZA ABOVE required?

We will look at this in more detail later.

## Using other characters to create new characters

- Ideas for orthographies...and what's wrong with them
- Using a hamza for a new character

*The general principle is that when such a hamza is used to indicate an actual glottal stop (or the /je/ sound used in Persian and Urdu for ezafe), it should be represented with a separate combining mark, either U+0654 arabic hamza above or U+0655 Arabic hamza below. However, when the hamza mark is used as a diacritic to derive a separate letter as an extension of the Arabic script, then the basic letter skeleton plus the hamza mark is represented by a single, precomposed character. (Chapter 9 of TUS)*

- When using U+0628 plus U+0654, if normalization is done, vowel marks have potential for coming between a character and the

hamza: هَ

2017 ©SIL International

Regarding the hamza, the Unicode Standard says “The general principle is that when such a hamza is used to indicate an actual glottal stop (or the /je/ sound used in Persian and Urdu for ezafe), it should be represented with a separate combining mark, either U+0654 arabic hamza above or U+0655 arabic hamza below.

Chapter 9 Middle East-I: Modern and Liturgical Scripts.  
<http://www.unicode.org/versions/latest/ch09.pdf>

## Using other characters to create new characters

- Ideas for orthographies...and what's wrong with them
- Using FBB2 - FBB9 ( )

2017 ©SIL International

Some new marks were added to Unicode recently. They are little one, two and three nukta (above and below). Some people have the idea we could use these to form new characters. These are not for forming new characters. They were added for pedagogical purposes and should not be used in orthographies. If you find you really need a new character that looks like something else with an added nukta or two or three, we might need to propose that character to Unicode.

Chapter 9 Middle East-I: Modern and Liturgical Scripts.  
<http://www.unicode.org/versions/latest/ch09.pdf>

## Using other characters to create new characters

- Ideas for orthographies...and what's wrong with them
  - Using Koranic marks to form a new character
    - 06DB (◌ْ◌ْ)
    - 06E2 (◌ْ◌ْ)

2017 ©SIL International

Some African orthographies have started using koranic marks to form new characters. Why is that a bad idea?

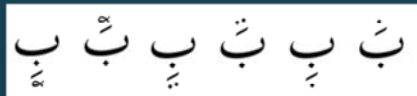
Koranic marks are there specifically for quranic purposes. In general, the expected behavior for a quranic mark is like an honorific. It may float a little above the main character (including vowels). If you use it to form a consonant, vowels will likely come between the base character and the quranic mark.

If you find you really need a new character that looks like something else with an added meem or noon above it, we might need to propose that character to Unicode.

Chapter 9 Middle East-I: Modern and Liturgical Scripts.  
<http://www.unicode.org/versions/latest/ch09.pdf>

## Using other characters to create new characters

- Ideas for orthographies...and what's wrong with them
  - What about tone?
  - Arabic language doesn't use tone
  - Rohingya example (U+08EA..U+08EF)



- Unicode Technical Committee said it was inappropriate to use Koranic mark U+06EC
- These tone marks are still not well supported in applications


2017 ©SIL International

Tone is a difficult thing because the Arabic language doesn't use tone and until recently no one was trying to use tone in Arabic script. We proposed some new tone characters for the Rohingya language. The request was for 3 tones. The tones should follow the vowel. So, if the vowel was below, then the tone needed to be below. If the vowel was above, then the tone needed to be above. Unicode required us to propose 6 characters to do that.

The tones are the outer one dot, two dots and fish looking characters.

In this case, we needed to make a Unicode proposal, but we are still not seeing good applications support for the characters as the language is not a high value to the industry.

## Using other characters to create new characters

- Ideas for orthographies...and what's wrong with them
  - Using Koranic marks for a different purpose – such as tone or +ATR
  - 06EA ()
  - Standardizing the order of Arabic combining marks ([link](#))

2017 ©SIL International

What about using Koranic marks for another purpose such as tone? Will there be unexpected ramifications? I don't really know. If the placement on the outside is what you want, then you'll probably get what you want. However, Unicode is working toward enforcing combining mark positions in Arabic script and I'm not sure what the long term results will be. I do know the UTC would not recommend using Koranic marks for something else. If you are doing that, we should consider making a Unicode proposal.

OR, we could ask the question to UTC about whether this is a valid use of the character. If they agree, then we should get them to document it and if they say we should propose a character we could do that.



## Using different codepoints

...depending on the position in the word

- Do not change underlying character depending on position in the word, the font should handle it as we see with the Farsi Yeh

Character	Final	Medial	Initial	Isolate
U+0649 Alef Makura	آ	ا	أ	آ
U+064A Yeh	ي	ی	ی	ي

- Problems caused: Searching, Modified words require changing codepoint

Some people have chosen to use different codepoints, depending on where in a word the character occurs. This is so they can get dots or no dots. There are characters that already do this for you. Farsi Yeh is a common example.



## Using different codepoints

...depending on the position in the word

- Do not change underlying character depending on position in the word, the font should handle it as we see with the Farsi Yeh

Character	Final	Medial	Initial	Isolate
U+0649 Alef Makura	ى	ـ	ا	ى
U+064A Yeh	ي	ي	ي	ي
U+06CC Farsi Yeh	ى	ي	ي	ى

- Problems caused: Searching, Modified words require changing codepoint

Some people have chosen to use different codepoints, depending on where in a word the character occurs. This is so they can get dots or no dots. There are characters that already do this for you. Farsi Yeh is a common example.

## Use different codepoints depending on the position in the word

Character	Final	Medial	Initial	Isolate
U+066f Dotless Qaf	ق	م	و	ق
U+06a7 Qaf with dot above	ق	ف	ف	ق
U+08BC African Qaf	ق	ف	ف	ق

2017 ©SIL International

Another example of a set of characters we proposed for West Africa is the African Qaf. People were using the dotless qaf in some positions and the qaf with dot above in other positions. The new African Qaf will give them what is needed, however it will be some time before applications support the new characters!

## Using hidden characters

- 200C ZERO WIDTH NON-JOINER (ZWNJ)
  - Not: میکنم
  - This: میکنم (insert U+200C after yeh)

The hidden characters may not be a part of your standard orthography, but it is important to be aware of them.

Here are some characters you might want to include in your list of characters required.

ZWNJ does precisely what it says. It causes characters to NOT join together such as in the case of the yeh.

## Using hidden characters

- 200D ZERO WIDTH JOINER (ZWJ)

ي ي ي

(U+064A\_ U+064A U+200D \_ U+200D U+064A U+200D \_ U+200D U+064A)

- لله (U+0644 U+064E U+0644 U+0647)
- لله (U+0644 U+064E U+200D U+0644 U+0647)
- nd (Africa)
  - ند (no sukun) -- U+0646 U+062F
  - نْ (sukun) -- U+0646 U+0652 U+062F

ZWJ does the opposite. It is not normally used in an orthography, but it can be useful to show the different forms a character can use.

## Using hidden characters

- 200F RIGHT-TO-LEFT MARK (RLM)
- 200E LEFT-TO-RIGHT MARK (LRM)
  - 56-34:12 (RLM before colon and before hyphen)
  - 34-56:12 (RLM before colon and LRM before hyphen)

Then we have right-to-left and left-to-right marks.

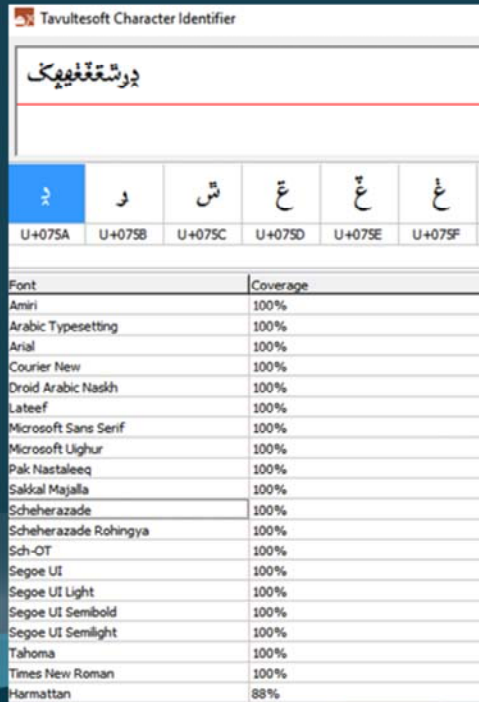
These can be used for cross references.

The last examples is not a normal use of reference, but we do know of one or two cases where that is the preferred reference.

Make sure to put these characters on your keyboard if you need them!

## What fonts support my characters?

- Tavultesoft Character Identifier (Windows)
  - <https://goo.gl/8wyfII>



Font	Coverage
Amiri	100%
Arabic Typesetting	100%
Arial	100%
Courier New	100%
Droid Arabic Naskh	100%
Lateef	100%
Microsoft Sans Serif	100%
Microsoft Uighur	100%
Pak Nastaleeq	100%
Sakkal Majalla	100%
Scheherazade	100%
Scheherazade Rohingya	100%
Sch-OT	100%
Segoe UI	100%
Segoe UI Light	100%
Segoe UI Semibold	100%
Segoe UI Semilight	100%
Tahoma	100%
Times New Roman	100%
Harmattan	88%

2017 ©SIL International

This has nothing to do with orthographies...except you want to know if there's a font that supports your new orthography.

The question of what fonts support my character can be difficult to come up with. There are various tools you can use. One easy starting point for Windows users is the Tavultesoft Character Identifier. You can paste your list of characters in there and see which fonts support your characters. If you click on the font, it will display those characters for you.

This tool is also useful if you have a string of text and you are not sure what the codepoints are.

Copy following text:

اُگلا پنتچہ غطصیجٹک

## What fonts support my characters?

- Character Set Inspector
  - <http://graphiccore.github.io/charset-inspector/>
  - Drop Character set file in left-hand box
  - Drop font in right-hand box

Go to website and demo.

## Unicode proposal

- Process for Unicode proposals
  - Examples of usage
  - Describe language using the character and how it is used
  - Defend why any similar characters would not be suitable
  - Give character properties
  - Give sort order
  - Give list of characters it could be confused with
- Expect the process to take years
- Expect application support to take years
- Might be worth it if the language community does not want to change their orthography

So, if we've decided we need to make a Unicode proposal, there are a number of things we need to do.