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

Open Tavultesoft Character Identifier
Open folder CharacterSetInspector
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+0268 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?
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.
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.
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.
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.
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.
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.
Why was ARABIC LETTER BEH WITH HAMZA ABOVE required?

We will look at this in more detail later.
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.

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: ﺏ
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.
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.
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 they 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.
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.
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.
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.

<table>
<thead>
<tr>
<th>Character</th>
<th>Final</th>
<th>Medial</th>
<th>Initial</th>
<th>Isolate</th>
</tr>
</thead>
<tbody>
<tr>
<td>U+0649 Alef Makura</td>
<td>ي ی ی</td>
<td>ب ب ی</td>
<td>ب ب ی</td>
<td>ب ب ی</td>
</tr>
<tr>
<td>U+064A Yeh</td>
<td>ي ی ی</td>
<td>ب ب ی</td>
<td>ب ب ی</td>
<td>ب ب ی</td>
</tr>
<tr>
<td>U+06CC Farsi Yeh</td>
<td>ي ی ی</td>
<td>ب ب ی</td>
<td>ب ب ی</td>
<td>ب ب ی</td>
</tr>
</tbody>
</table>

- Problems caused: Searching, Modified words require changing codepoint
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!

<table>
<thead>
<tr>
<th>Character</th>
<th>Final</th>
<th>Medial</th>
<th>Initial</th>
<th>Isolate</th>
</tr>
</thead>
<tbody>
<tr>
<td>U+066f Dotless Qaf</td>
<td>ق  و  م  ت</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U+06a7 Qaf with dot above</td>
<td>ق  ف  ن  خ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>U+08BC African Qaf</td>
<td>ق  ف  ن  خ</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.
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.
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 prefered reference.

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

لاّ وَأَمْلِي چَرُّشَشَنَّ
What fonts support my characters?

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

Go to website and demo.
So, if we’ve decided we need to make a Unicode proposal, there are a number of things we need to do.