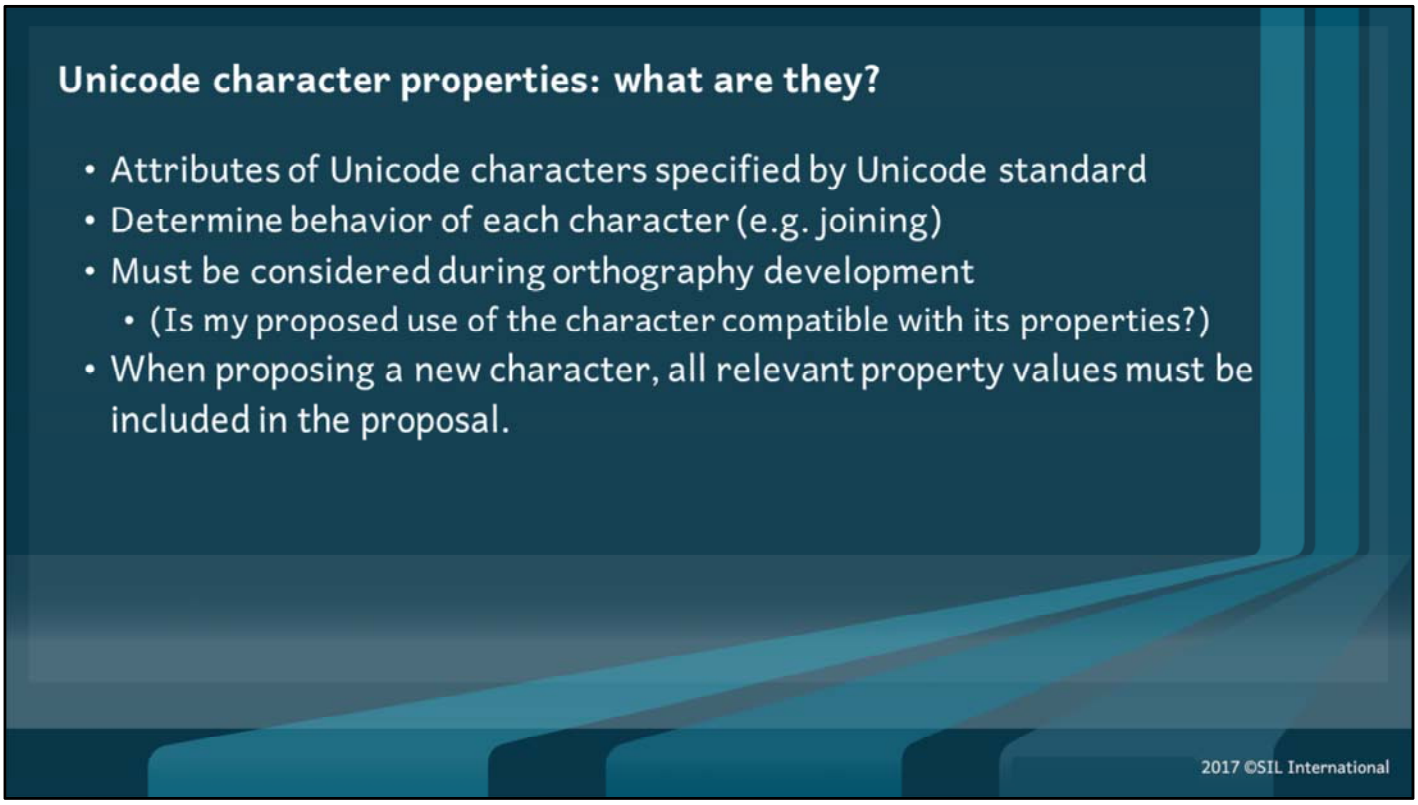
Unicode character properties



This will be a brief review of some of the character properties that are important for Arabic characters and which will need to be defined for any proposed new characters.

Unicode character properties: Where to find them

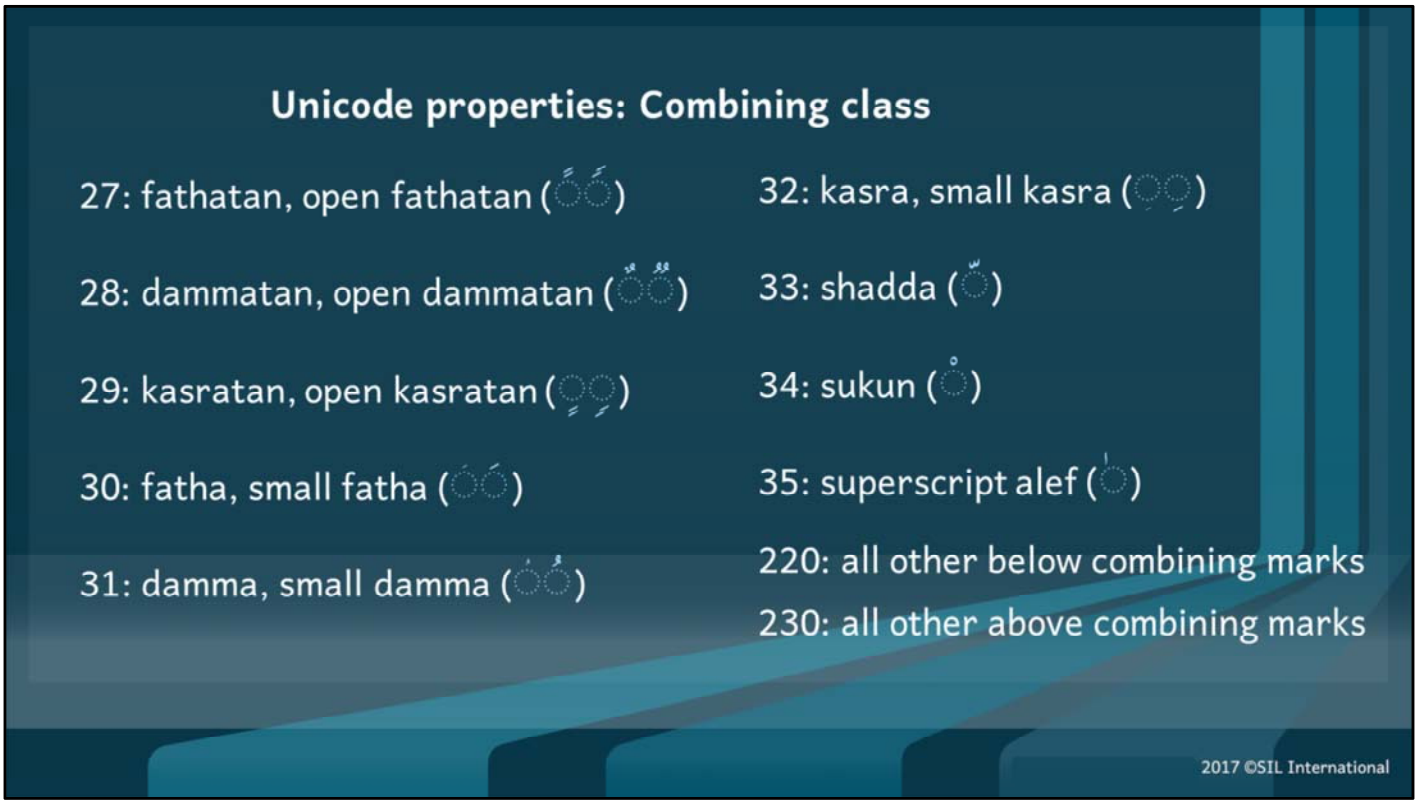
- The Unicode Character Database (UCD) data files
  - Plain-text files (but with strict formatting rules)
    - Line-oriented and structured XML variants are available
  - http://www.unicode.org/Public/UCD/latest/

- Additional ways to view properties
  - Richard Ishida's UniView webapp: https://r12a.github.io/uniview/
  - Unicode properties Excel workbook: http://scripts.sil.org/excelunicodedata
    - Currently at v 7.0
  - Unicode's Unibook windows app: http://unicode.org/unibook/
  -
  -
  -

Unicode prescribes every character's "Joining type" and "Joining group". These are all defined in a file called "ArabicShaping.txt". It's important to understand this so that you will know the expected behavior for a characters.

The Joining type describes how a character joins. An example of Right joining is an Alef. It only joins on the right. And example of a Dual joining character is a seen or a beh. As you know, these join on the left or the right. The only Join Causing character that we are likely to need is U+200D. A number of Arabic characters are in the Non Joining type. An example of this would be the End of Ayah. This doesn't mean it doesn't require special rendering, it just means it doesn't actually link with another character.

Next we have Joining Groups. This defines what kind of joining behavior to expect. If a character is in the BEH group, then we would expect it to behave as the standard "Beh" behaves.



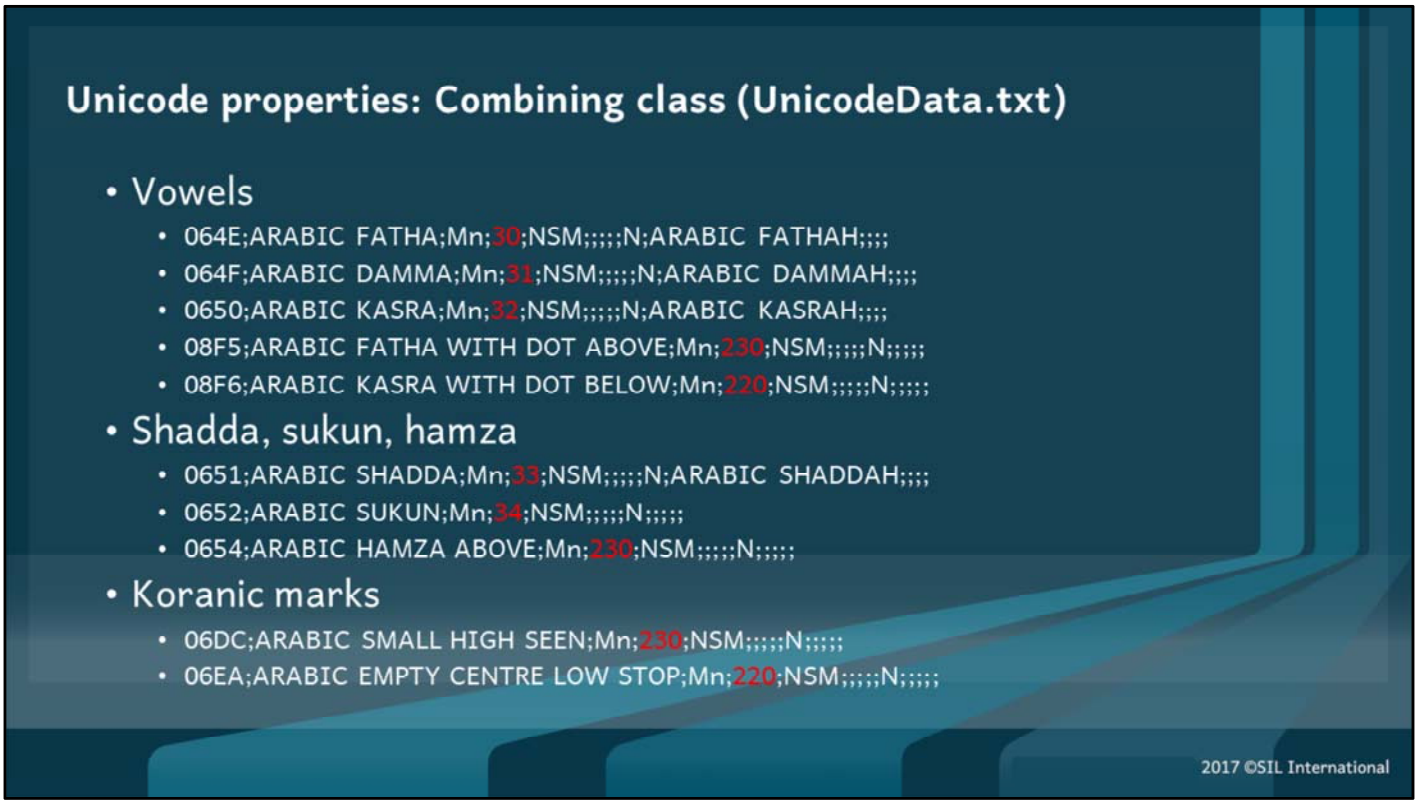
The combining class value determines how combining marks are reordered in your text during Unicode normalization (most often used for string storage and comparison).

Can anyone suggest why this question is important during orthography design?

Ans: Suppose you have diacritics from two different classes, say a shadda (cc=33) and damma (cc=31), on the same base letter. Text processes are free to change the relative order of the diacritics. In particular, Unicode normalization will always put the damma before the shadda no matter what order they were in the original text. This means – from an orthography design perspective – that one *must not make meaning distinctions* between, damma followed by shadda and shadda followed by damma -- searching, sorting, comparison should all treat either order as meaning the same thing. (Also, font designers should render damma+shadda identically to shadda+damma, but not all fonts do this correctly).

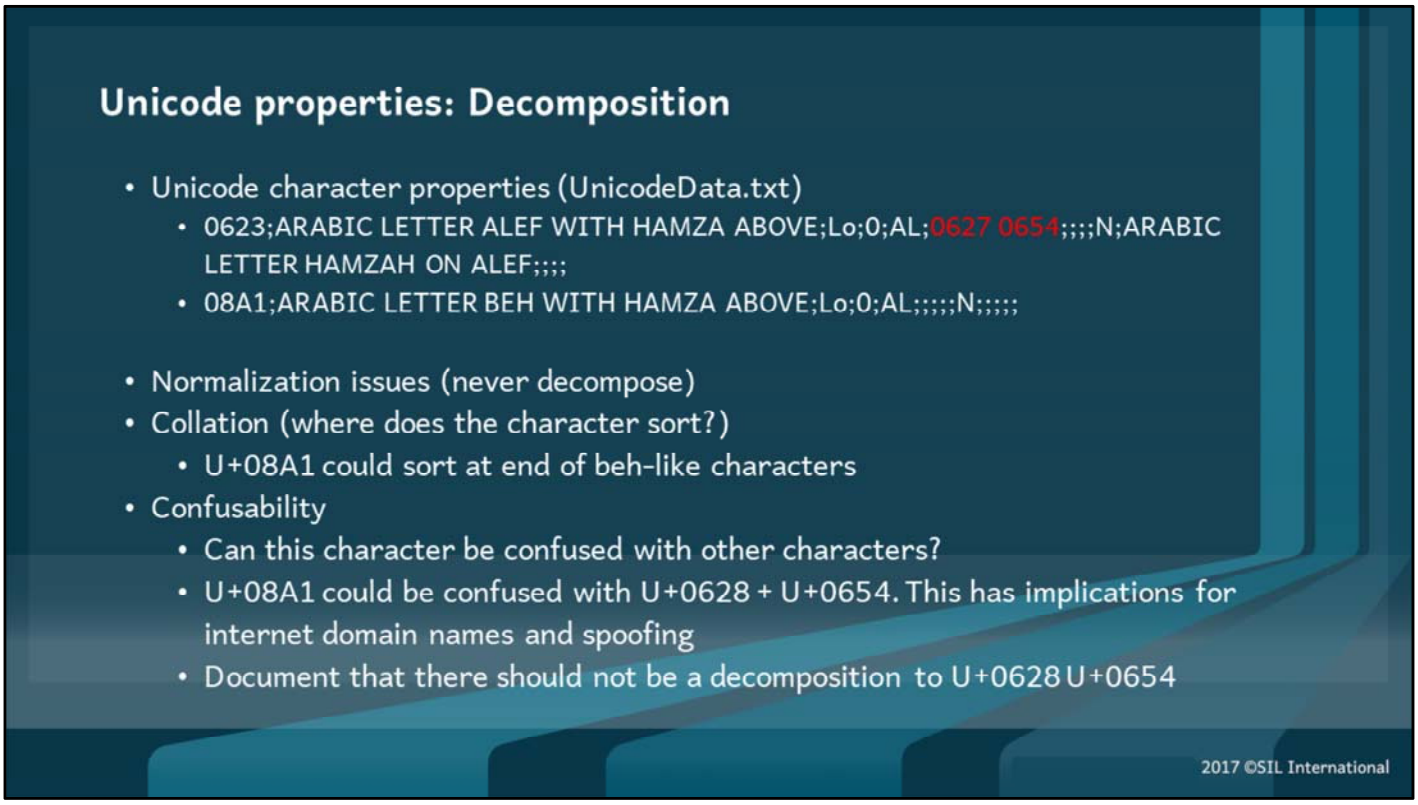
Does anyone see anything illogical in these combining class values?

Ans: Do you think it is logical for a fatha or damma to precede shadda?



Extending this to more marks: if you have a shadda (cc=33), fatha (cc=30) and a koranic mark above (cc=230), from a technical point of view, these mean the same thing no matter which of the 6 possible orders they are in your text!

If there are multiple combining marks *from the same class*, say from cc=220, Unicode normalization will not change the relative order of these marks in your text. While this potentially means that you can treat different orders as having different meanings, extreme caution is needed as there are other rules at play (such as for the koranic and tone marks)



Another important Unicode character property is Decomposition. An example is 0623 which is an alef with a Hamza above. Unicode character properties declare that this character *is the same thing as* the two character sequence alef plus combining Hamza. From the perspective of what does the text mean, there is no difference between 0623 and the sequence 0627+0654.

This is common in Latin script, right? Someone give me an example?

Recalling the earlier example (in the previous presentation) of the new character beh+Hamza – note that the decomposition field is empty indicating there is no decomposition.

Ok, does decomposition need to be taken into account when making orthography decisions?

[discuss]

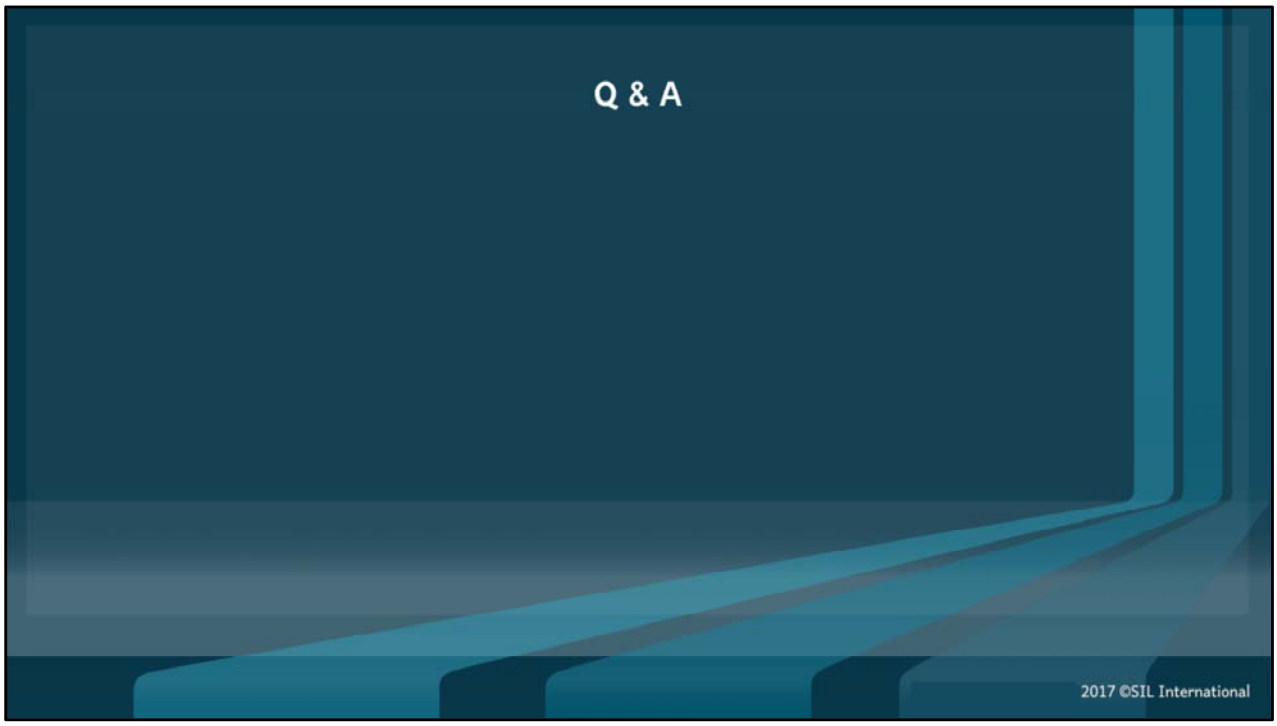
In the Unicode proposal a number of questions have to be answered, including:

- impact on normalization (we already noted there is no decomposition so it has no impact on normalization)
- Collation: we have to specify a default sort order for this character relevant to

existing characters
- Confusibility

Unicode properties: Decompositions

Does this have a bearing on orthography decisions?


Q & A
2017 ©SIL International