Subject: Re: Choosing the best way to go full UNICODE
Posted by cbpporter on Wed, 31 May 2017 11:43:50 GMT
View Forum Message <> Reply to Message

Without RLE, I think you can get around this by positions not representing characters, but sequence starts.

As a mandatory condition, every time the position is updated, it is guaranteed to be at the start of a sequence.

As a more complicated example, let's say you have a selection of text, with a "begin" and "end" pos. You handle a key press. Taking your start pos as a sequence start, you determine the sequence end. This means looking to see how many code units it is, seeking over combination marks and ligatures. Basically on the fly glyph analysis. With sequence start end end you know for a fact that everything between these two values must go. You do the same for the end position. As an optimization, you can mark everything for deletion between the start sequence begin and end sequence end. The text marked to be replaced will replaced with multiple code units.

The real challenge is to standardize these operations so you don't have to repeat them.

Maybe we need some GlyphInfoExtractor class or something. Something when given a random sequence of code units and a valid code point start, it can handle such common operations?

Here is a sample from unciode.org:

This is 14 code units, 5 code units, 4 glyphs. The user will see and recognize 4 items as more or less "atomic", so we should focus on this.

We need and API that can locate each glyph start and allow us to replace glyphs 2 and 3 with an

This can be done on the fly with something high level like:

or

or we can go lower level. Or we can go into multi-byte String territory.

PS: the high level stuff still is StringWalker territory.

File Attachments

1) char_combmark_ex1.png, downloaded 862 times