L2/16-315R

Re: **Handling interaction between UCD and emoji properties**
To: **UTC**
From: **Mark Davis, Peter Edberg**
Date: **2016-10-31**
Draft: **https://goo.gl/8bdlsG**

We have the following action from UTC #148:

148	**A006**	**Mark Davis, Peter Edberg, Emoji Subcommittee**		**Produce a proposal for handling the interaction between segmentation and emoji properties. Either (1) move some properties into the UCD, or (2) decouple properties from segmentation. Spell out the preferred option(s) and alternatives.**

There are two areas where the emoji properties interact with the UCD properties, in particular, where changing the value of the property Emoji=No to Emoji=Yes should cause other properties to change.

1. Segmentation
    - The Grapheme Break, Word, and Linebreak properties have values that depend on the Emoji property
2. Variation Selectors
    - The UTC has committed to maintaining the invariant that if a character has the property value Emoji_Presentation=No, then it has Emoji and Text variation selectors (VS16 and VS15). Those are established by the data in StandardizedVariants.txt

There are alternatives for dealing with the current implementation issues resulting from the async development of UTC properties and emoji properties. Option C is the recommended approach at this point. This leaves open the option A once emoji properties only need to change on an annual cycle.

A. Move some emoji properties into the UCD.
    - This is the simplest approach, but constrains those emoji properties to be only changed with yearly (June) Unicode releases, when (for now) emoji are developing on a faster pace. It also makes the Unicode release yet larger and more complicated.

B. Decouple emoji properties from UCD properties
    - UCD algorithms only depend on UCD properties, but are broadened to deal with “future” emoji.

C. Use TR51 and CLDR as the vehicle
    - CLDR (Current CLDR/ICU Segmentation) is used for customizing segmentation to deal with emoji (as is current).
        i. We are clearer in the text for segmentation (#14, #29) that it references the current release of tr51.
        ii. Documenting as we do now that CLDR customization rules are recommended.
    - We change the definition of allowed variation sequences to also redirect to a data file that is part of TR51.

Variation Selectors

Decoupling
    Grapheme Break, Word, and Linebreak

Current CLDR/ICU Segmentation

Variation Selectors

The other problem for changes to emoji properties are the StandardizedVariants.txt. There are a few ways to decouple these.

The simplest way is to define that the valid sequences with VS15 and VS16 are no longer established solely by the presence of those sequences in StandardizedVariants.txt. We already defer to IVD for some of the variation sequences,

and we would defer to the EVD (in tr51). We then move the items with VS15 and VS16 from StandardizedVariants.txt to (provisional name) http://unicode.org/Public/emoji/XX/EmojiStandardizedVariants.txt.

We document in TR51 the valid emoji variation sequences to be everything in EmojiStandardizedVariants.txt, which can be a file derived from other emoji properties. The definition ***ED-9. emoji variation sequence*** would be modified as necessary and point to the list in the new file.

In Unicode 10.0 we document this "deferment" for emoji standardized variants, and add info in the header of StandardizedVariants.txt.

Decoupling

The following presents a possible approach to decoupling. The advantage of decoupling is that it allows for (limited) changes to emoji at a different pace than the annual Unicode release. These do not include new characters, of course, but do allow for additions of ZWJ and modifier sequences that can involve changing Emoji and Emoji_Modifier_Base property values, and enabling certain variation sequences. Eventually, these properties should be included in the UCD, but it is slightly premature to do it now.

The UTC had already decided to "future proof" the segmentation of emoji, by providing data that removes segmentation boundaries between "emoji-like" characters that could possibly get the value Emoji=Yes in the future, and be part of ZWJ sequences. Because there was not enough time before the Unicode 9.0 release (June 2016), this was done in CLDR 30 (Oct 2016). It takes the form of a CLDR property, plus customized segmentation rules. Those are listed at the end of this document, and already implemented in ICU and other libraries; these do also require overriding StandardizedVariant.txt.

Grapheme Break, Word, and Linebreak

We could decouple the Emoji property from segmentation by creating a new **UCD** property called **Extended_Pictographic (EP)** with the contents being the CLDR values for (Extended_Pictographic | EmojiNRK). The amended segmentation rules in UAX 29 and 14 would become:

> GB11′ Extended_Pictographic ZWJ × Extended_Pictographic
> WB3c′ ZWJ × Extended_Pictographic
> LB8a′ ZWJ × (ID | Extended_Pictographic)

The draft contents for this can be taken from the current Current CLDR/ICU Segmentation, then refined based on feedback from the membership and public.

Note: the name and contents of this property are up for discussion.

That allows any of the Extended_Pictographic characters (like the MALE SIGN) to be changed to have Emoji=Yes without affecting segmentation.

Current CLDR/ICU Segmentation

- Property: ExtendedPictographic.txt
- Customized Rules: LDML: Extended Pictographic

The relevant text from the from the LDML spec for customizing the rules is:

> Let Extended_Pictographic be defined as in ExtendedPictographic.txt
> Let EmojiRK = [\p{GCB=Regional_Indicator}[*#0-9©®™〰〽]]
> Let EmojiNRK = [\p{Emoji=Yes}-EmojiRK]
>
> The customized rules replacing GB11, WB3c, and LB8a are:

GB11′ **(Extended_Pictographic | EmojiNRK)** ZWJ × **(Extended_Pictographic | EmojiNRK)**
WB3c′ ZWJ × **(Extended_Pictographic | EmojiNRK)**
LB8a′ ZWJ × (ID | **Extended_Pictographic | EmojiNRK**)

The future-proofing of the segmentation rules handles the change from Emoji=No to Emoji=Yes by having an expanded set of characters that could possibly have their Emoji status changed in that way, so it effectively decouples the two properties. That is, as long as characters changed from Emoji=No to Emoji=Yes are in Extended_Pictographic, the UCD does not need to be changed in order for segmentation to still work. Of course, if any characters outside of Extended_Pictographic would need to be changed, that would just have to wait for the next version of Unicode.

That means that Extended_Pictographic needs to (a) include all the prospective characters, and (b) not include extraneous characters (that are neither pictographic symbols nor emoji-like).