L2/17-375

Re: **Remaining Tag Sequences**
From: **Mark Davis**
Date: **2017-10-13**
To: **UTC**

The TAG sequences we introduced in Emoji 5.0 have been successfully implemented by major applications. When we finalized them, we held back on possible applications, focusing on the main use case. (The concept had been introduced earlier, but went through many iterations before we were satisfied with the design.)

The original discussion included:

1. Gender
2. Hair
3. Direction
4. Private Use

We ended up deciding that there are better mechanisms for #1 and #2. However, we never came to agreement on the other two. This document discusses the other two cases. It recommends that for Direction we use a ZWJ sequence approach instead of tags. On the Private Use tags, it simply refreshes recaps the discussion, using the new syntax, but doesn't take a position.

Direction

The fact that emoji sometimes face "the wrong way" is a continuing source of complaints. As discussed in Jane Solomon's eLex Keynote, direction can have an effect on the meaning. Adding a mechanism to change direction does have the nice feature that it is possible to implement without much impact on the size of the emoji font, using the graphic system to flip images horizontally.

The original document considered addressing this by having tag sequences that would be equivalent to:

tag_base	RUNNER, PISTOL, … (except enclosed alphanumerics)	
tag_spec	Dr✦	Right-facing
	Dl✦	Left-facing

An alternative solution is instead having ZWJ sequences of the following form. There are two alternatives, because vendors can differ as to the default direction.

Appears to User	Internal Representation	Fallback Appearance
🏃	🏃 ZWJ ⬅️	🏃 ⬅️
🏃	🏃 ZWJ ➡️	🏃 ➡️

As with gender, the advantage of a ZWJ sequence is more obvious fallback on older systems. A tag sequence may not show a different appearance than just the base (although a different appearance is recommended), whereas a ZWJ sequence is clearly something recognizably different.

A small number of such sequences could be added to RGI for the most obvious cases, and others could use the normal RGI proposal process.

Originally, we had that:

> The directions are to have a mirrored effect in a bidi context. All emoji characters are Bidi_Class=Other_Neutral (except for the enclosed alphanumerics).

That complication is probably not be worth retaining, since it less unlikely that the writer and intended readers of sequences of emoji would typically have the same bidi contexts, including implicit contexts determined by adjacent characters.

Private Use

There has been some call for a mechanism that allows for customized emoji (such as Coded Hashes of Arbitrary Images), though none that have addressed the full lifecycle. That is, how a program would lookup an image from a customized sequence, how to address the troubling security implications in plain text.

A Private Use tag sequence would allow for experimentation with approaches to this to see if something feasible could be devised. While it is not clear whether this is a good idea or not, the following updates the proposal to today's syntax:

> Private Use tag sequences are for closed interchange within a given system. As with private use codes in general, the tag sequence may have no meaning or a different meaning outside that system, so it is not suitable for general interchange.
>
> tag_base [:emoji:]
> tag_spec X[[\x{E0020}-\x{E007E}]-[A-Z]]+✦ Private use customization
>
> Example: 🏃 Xsweating✦
>
> Implementations should consider the use of any of the thousands of *private use* Unicode characters instead.
>
> However, the advantages of Private Use emoji tag sequences include:
>
> - There is a fallback to display of the base emoji character, instead of showing a black box
> - The sequences would be recognized as emoji by any system, whereas a *private use* Unicode characters would not.
> - The private use Unicode characters could accidentally collide: the same character used by different vendors could mean different things.

That formulation still has the disadvantage that the private use tag sequences could still accidentally

collide. One variant that we discussed back in 2015 would be an initial domain name in the spec. That would have the additional advantage that:

- Vendors could be assured that their customizations would not accidentally collide with other vendors.

If we took that route, the tag_spec would look like the following.

tag_spec	X[A-Z]+[[\x{E0020}-\x{E007E}]-[A-Z]]+✦ where [A-Z]+ is interpreted as a domain name.	Private use customization
Example:	🏃 XIBM.COMabcd✦	

Note that short domain names would lower the overall storage, and allow more room for a private designator, since the overall number of codepoints for base + spec is less than 33. We could possibly assume ".COM" if there is no dot, to make the storage smaller.