

Re: Unicode Customized Emoji (UCE) Proposal
Date: 2015-10-21 (aka BTTF day)
From: Peter Edberg, Mark Davis, and the emoji SC
Draft: <https://goo.gl/ocVooV>

The following document was developed at the request of the UTC, based on the discussion in the UTC at the last meeting. *It is only a proposal for discussion, and must not be deployed until and if it is finalized and approved.*

Motivation

There is significant demand for additional “customized emoji” that work like existing emoji, and are available in a more timely fashion. Requests for these have been coming in to the Unicode Consortium (as well as to some member organizations), but these emoji requests are not well-suited to the traditional encoding process.

The UTC does not want to devote the bandwidth to handle them, nor can the traditional encoding process respond as quickly as desired for the emoji requests. Furthermore, some users/implementers might want to use/support some pictograph that is not appropriate for standardization in Unicode. e.g., it’s likely to have short-term-only usage.

Scenarios

Customized emoji should support usages such as the following:

1. Flag pictographs
 - Flags of several regional subdivisions such as England, Wales, Scotland
 - Flags of several supra-national organizations such as the EU and the UN
 - Certain other popular flags such as the skull & crossbones
2. More specific appearances for human figures such as WOMAN or MAN, e.g.
 - hair styles and color
 - eyeglasses
 - facial hair
3. Emoji for different breeds of dogs, or for variations of a MONSTER (vampire, zombie, werewolf,...)
4. Pictographs whose lifetime is expected to be relatively short, e.g.
 - Caricatures of political candidates during an election season.
 - Images associated with Internet memes

High-level goal

To enable use of customized emoji in interoperable text interchange, with processes that allow for rapid innovation and responsiveness to user demand for additional emoji.

Customized emoji would successful only when

1. the process for establishing new customized emoji allows for
 - a. a larger number of emoji being defined per year, and
 - b. less overall involvement of the Unicode Consortium
2. and the customized emoji remain interoperable, meaning that
 - a. they are stable over time, and across platforms
 - b. that is, they maintain the same core “visual semantic” (while allowing some variation)

Specific goals

Mechanism and process goals for users and vendors:

1. Customized emoji work like existing emoji
 - They can be input from “normal” keyboards/palettes
 - They can be used in e-mail subject lines, SMS messages, filenames
 - They scale with text
2. Interoperability issues across platforms, system versions
 - It should be possible for any vendor to implement any customized emoji. That is, there is an easy, mechanical way to find out exactly what all of the customized emoji are at any point in time, to sufficient detail to be able to implement them.
 - For customized emoji that are not supported on a given platform or system version, there should be a fallback display that provides some useful information about the identity of the customized emoji
 - The number of different customized emoji is tractable; not so large that it would be a burden to support all or most of them.
3. It should be possible for implementers to support customized emoji in various application contexts or protocols without significant change in security considerations associated with that context or protocol.
4. Customized emoji can be created with relatively quick turnaround
5. Stability: The mechanism will representing customized emoji will not disappear, and the interpretation of a given customized emoji will not change radically over time
 - For example, an emoji representing a particular person will never reassigned to represent a different person
6. Customized emoji can be “deprecated” — there should be a way to indicate that customized emoji are past their useful life.

Mechanism and process goals for UTC:

1. Customized emoji can be created with minimal impact to the UTC
2. Customized emoji creation need not be tied to Unicode releases
3. Looser criteria apply for the creation of customized emoji than do for the creation of normal emoji
4. Since customized emoji will be variants of existing emoji, the UTC would focus more on different kinds of “generic” emoji that can serve as a base for customized emoji.

Technical Overview

1. Each customized emoji should involve a small number of Unicode characters (we are currently considering a limit of 10 or fewer).
2. Each new emoji should consist of a standard Unicode character that provides a fallback appearance (with some indication of the type of emoji sent), plus additional characters that provide the more specific appearance on systems that support new emoji but will be ignored by older systems
 - Implementations that do not support the mechanism at all may display just the fallback character
 - Implementations that support the mechanism but do not support a particular sequence should fall back to display of that standard emoji character plus some indication that there is additional undisplayed information
 - This also implies that a consideration for encoding of standard emoji is their potential usefulness as fallbacks for customized emoji.
3. Each customized emoji sequence should be treated as a cluster for segmentation purposes (if possible without changing UAX 29, so they work correctly on older systems)
4. Each customized emoji sequence should have defined Unicode properties, based on the base emoji.
5. Requirements that derive from security considerations
 - Rendering an emoji (at least the first time) may involved a request for image data from an server. (Similar to an `` in HTML.)
 - We may want to restrict this in some way.
6. The customized emoji mechanism must be implementable, feasible, and well documented
7. The UTC can define its own customized emoji, and can also define a process that permits customized emoji to be defined by reference to external sets of coded entities.
 - An example of the latter is already provided by the `unicode_region` flag emoji, and by the proposed `unicode_subdivision` flag emoji.
 - This implies that there must be a mechanism to identify the authority for an “external set” of an customized emoji.
 - The UTC would need to develop a set of criteria for recognizing such an authority, so that the proper standards for reliability, stability, and accessibility are observed.

Basic mechanism

The proposed mechanism uses a sequence of characters:

1. It begins with a standard Unicode emoji character, called the base (more details below)
 - a. This will be used as part of the fallback mechanism and should provide an indication of the general category of the customized emoji (e.g. flag, emoticon face, etc.).

2. That is followed by a sequence of TAG characters in the range E0020..E007E (details below)
 - a. Open issue: Need to define maximum number of TAG characters, such as 8.
 - b. Open issue: Should one of these (maybe E007E?) be reserved as a terminator character that must end the sequence? This makes font ligatures somewhat more useful, and also processing text; but costs an extra character.

Tag sequence structure and details

1. The initial TAG characters, in combination with the base character, designate the “namespace”, which determines the mechanism used for the remaining non-terminator tag characters. The namespaces and corresponding mechanisms are assigned by UTC.
 - a. Sometimes namespace includes *any* base character; in other cases it may be restricted to certain base characters.
 - b. Some namespaces will have 1 TAG character. If those start to run out, then the remaining ones will be reserved for being initial parts of longer sequences.
2. Base <any> + Tag U is reserved for the Unicode Consortium itself. There are two initial uses
 - a. base U+1F3F3 WAVING WHITE FLAG + TAG U,
 - i. the remaining tag characters are a specification of either a valid Unicode [subdivision attribute](#) or a valid 3-digit unicode_region_subtag. They are in canonical form, so no uppercase.
 - ii. This can designate a pictograph for whatever is currently the flag of the specified subregion.
 - iii. It is not intended to provide a mechanism for stable representation of any particular flag image.
 - iv. This delegates decisions about regional subdivisions to another authority (based on ISO), which can avoid a lot of work and also remove perceptions of bias on the part of Unicode.
 - b. base U+1F3F3 WAVING WHITE FLAG + TAG F,
 - i. the remaining tag characters start with TAG F, and are followed with one or more TAG digit characters that specify a number in a Unicode registry of pictographs for flags.
 - ii. This could handle e.g. UN flag, pirate flags.
 - iii. This does provide a stable representation for a particular type of image. The degree of variation is similar to that for existing emoji characters.
 - c. *(For discussion) We have gotten other requests for “transforms” that could use this mechanism.*
 - i. *Hair color (eg ginger, blonde, brunette, black, gray, white)*
 - ii. *Point right vs Point left (for faces with direction, hand gestures, vehicles, pistol, etc.)*
 - iii. *Gender (make a runner be male vs female)*

- iv. *For each of these, one would need two variants, because the base character could be either variant (or a neutral form).*
- v. *ZWJ sequences*
 - 1. *Glasses need not be included, since they could be done with a ZWJ sequence.*
 - 2. *There have been requests for adding beards, such as to man with turban. Instead of a transform, we could add “BEARD” and then use ZWJ.*
- 3. Example external authority
 - a. Base <any> + TAG A,
 - b. the remaining tag characters are specified in a registry maintained by organization X, authorized by the UTC.
- 4. Private Use
 - a. Base <any> + TAG X
 - b. the remaining tag characters are arbitrary, a Private Use customization.
 - c. This is subject to the usual limitations of private use: it only works well within a restricted domain, because otherwise the sequences used by different organizations can collide. Best practice is to use a longer sequence that is unlikely to collide with others.

Interaction of this mechanism with other mechanisms

- 1. With respect to emoji modifiers and ZWJ sequences, a customized emoji sequence (base + TAG sequence) behaves as a single emoji character; an emoji modifier that affects the customized emoji should follow the complete sequence representing the customized emoji, and in a sequence such as <Char ZWJ Char TAG+> the TAG sequence applies only to the second character in the sequence.
- 2. Open Issue: would it be simpler to restrict this so that the TAG characters can only be at the end of an emoji sequence?

Process for customized emoji

- 1. Initially, all Unicode customized emoji would be processed by the emoji subcommittee (ESC), and candidates are posted for public review on a new page. Based on the results of public review, a recommendation is then presented to the UTC for approval (or the UTC could delegate that approval to the ESC).
 - a. A major difference from the current process for regular emoji is that once those are approved by the UTC, they can be published after the meeting and are immediately usable by vendors. So the process could be as short as a quarter, rather than a delay of 1.5+ years.
 - b. It would, however, have the same core restriction; the customized emoji would have to be reasonable variants of an existing base emoji character.

2. The UTC can also define a sequence by reference to another authority
 - a. We need to move slowly and carefully on this, to make sure that we have the proper processes in place for accessibility to the defining mechanism, and for stability and interoperability.
3. As we gain more experience, we can tune these processes, of course.