

Unicode Customized Emoji Framework

To: UTC
From: Peter Edberg, Mark Davis, Emoji Subcommittee
Date: 2016-01-19

Contents

- [Relationship among related documents](#)
- [Motivation](#)
- [Scenarios](#)
- [High-level goal](#)
- [Specific goals](#)
- [Technical Overview](#)
- [Basic mechanism](#)
- [Tag sequence structure and details](#)
- [Interaction of this mechanism with other mechanisms](#)

Relationship among related documents

The following three documents are intended to supplant [L2/15-252](#) (“Unicode Customized Emoji (UCE) Proposal”, 2015-10-21), as well as [L2/15-190](#) (“PRI #299 Background: Representing Additional Types of Flags”):

1. [L2/16-009](#) (this document, Unicode Customized Emoji Framework) describes the current understanding of the overall motivation, goals and framework for Unicode Customized Emoji (UCE).
2. [L2/16-008](#) (Unicode-Specified Emoji Customizations) describes three specific tag formats within that framework: U and V tags specified by the Unicode Consortium, and X tags for private use. These may be approved separately from approval of the overall framework, and are primary focus for the current meeting.
3. [L2/16-010](#) (Customized Emoji Tag Registration) discusses ideas for a future tag registry mechanism; this is the most speculative of the documents.

Motivation

There is significant demand for additional “customized emoji” that behave like existing emoji, but 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 is likely to have short-term-only usage.

Scenarios

Customized emoji could 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.
 - Specific hair styles and color.
 - Presence of eyeglasses or facial hair (though the ZWJ mechanism may be better-suited for this).
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 are displayed inline with text, and scale with text (that is, the size of the displayed image depends on the text point size).
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, in 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.
 7. The customized emoji mechanism should be capable of being supported relatively easily by existing font rendering mechanisms, and should permit different levels of a text stack to function independently.

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 relatively small number of Unicode characters; the current proposal is for a maximum of 16 total characters.
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 a server (Similar to an in HTML).
 - Note: 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 provided by the `unicode_subdivision` flag emoji proposed as part of the Unicode-specified emoji customizations.
 - 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).
 - 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 U+E0020..E007E (details below)
 - The sequence is terminated by U+E007E TAG TILDE.
 - While not strictly needed (since the sequence ends when a non-TAG character is encountered), this can simplify font processing and minimize lookahead. Otherwise we would need to impose requirements such as that no tag string in the registry is a prefix of any other, etc.
 - To simplify implementations, we should define the maximum valid number of TAG characters, tentatively 15 including terminator.

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 a namespace includes *any* base character; in other cases it may be restricted to certain base characters.
 - b. Initial 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. Unicode Consortium will reserve and define certain namespaces and sequences, initially Base <any> + Tag U and Base <any> + Tag V; for details see [L2/16-008](#) (Unicode-Specified Emoji Customizations).
3. Base <any> + Tag X is reserved for private use; again, for details see [L2/16-008](#) (Unicode-Specified Emoji Customizations).
4. 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. For discussion of possible registration procedures, see [L2/16-010](#) (Customized Emoji Tag Registration).

Interaction of this mechanism with other mechanisms

Information about the use and interaction of customized emoji sequences with other mechanisms such as emoji modifiers and emoji ZWJ sequences is provided in the Overall Syntax section of [L2/16-008](#) (Unicode-Specified Emoji Customizations).