Re: **QID Emoji Proposal**

From: Mark Davis

Date: 2019-05-02 (last revised)

Proposal

Add a new TAG sequence definition using Wikidata QIDs to the Proposed Update of UTS #51.

The main purpose of this new format is to allow vendors, large and small, to choose, produce and interchangeable emoji TAG sequences that have well-defined denotations, but are not RGI. These can be supported in various applications without waiting for emoji to be encoded by Unicode.

By adding an initial proposal to the Proposed Update, we can then get feedback on the proposal in the 2019Q3 UTC meeting, and consider including it in version 13.0 of UTS #51. We would present two alternative options for the base character in review feedback: whether to allow different bases (for better fallback), or have a single base (to reduce alternative usages).

If incorporated, we would amend our processes to allow us:

1. to add such sequences to the set of RGI emoji sequences, as long as they satisfy the criteria in [Submitting Emoji Proposals](#) and have demonstrable high frequency of usage.
2. to add regular emoji characters using the normal process in [Submitting Emoji Proposals](#), using the demonstrable high frequency of usage of the QID emoji as evidence of potential frequency.

We may consider modifying our process to be able to add RGI sequences on a somewhat higher frequency than the current annual schedule, but that is not implied by this proposal.

Background

For many reasons, the introduction of emoji characters has been beneficial to Unicode and the languages that it serves. For example, because emoji support requires more complex features and supplementary code points, emoji implementations have also helped improve support of complex scripts used by billions of people around the world.

But the standard emoji characters are a closed set. Suppose that someone wants a *White-crested tiger heron* emoji or a *Raclette* emoji. Right now they go through a process of writing a proposal for a new character, and providing evidence of expected usage and other criteria. Because Unicode is limited to a relatively small number of new emoji per year, we normally require the expected usage to be above median (ie, roughly the frequency of the hamburger or of the Swiss flag).

While it is possible for a platform to support ZWJ sequences that are valid (but not [RGI](#)), the same is not true of single characters. A platform can use private-use characters (there are well over 100,000 Unicode private-use (PU) characters available) but those have major problems:

A. ***Collisions:*** there isn't anything to prevent two different platforms from using the same PU character for different things (collisions).
B. ***Behavior:*** an implementation can't determine which PU characters are intended to [*behave*](#) as emoji (requiring modification of property property mapping data for algorithms to support that behavior).
C. ***Discoverability***: an implementation can't determine which emoji a PU character is

supposed to be.

An alternative has been suggested using a hash-code approach for arbitrary emoji, but there are major problems with that that are as yet unaddressed. While collisions and behavior are not a material problem with this approach, discoverability is still a major issue. We have also considered over the years various approaches to making valid private use Emoji Tag Sequences, but those also run into the problem of discoverability.

But there is a possible mechanism that would handle the discoverability issue: a very large number of entities would have an automatic emoji definition, would prevent collisions, could be interchanged, would be discoverable, and could (where suitable) be integrated into RGI.

Internal code	Name	Sample Image
🥃Q673012✦	*Pisco Sour*	

For the public, this could result in faster deployment of emoji. For platforms, it would again allow faster deployment, and more freedom to experiment to see which emoji could be more frequently used. For Unicode, it would allow actual usage statistics rather than using proxies for expected frequency in emoji proposals. The potential downsides are primarily in interoperability, as discussed below.

Contents

QID Emoji Tag Sequences

This proposed mechanism defines a new type of Emoji Tag Sequence that uses Wikidata QIDs. These

are IDs that represent a Wikidata "item" entry — for more information, see Wikidata QIDs below. The mechanism would provide for a well-defined way for implementations to have the choice to support additional valid emoji, without waiting for Unicode to encode them.

Emoji tag sequences should be supported by any implementation that supports Emoji 5.0 (released on 2017-05-18). Older implementations would see just a fallback image.

Format

Technically, here's how this would work (I'll discuss Process later.)

We add a new emoji character EMOJI TAG BASE U+XXXX to Unicode 13.0.

We make a small addition to the formal specification: add the following option to UTS #51's valid emoji tag sequences. See **ED-14a**. *emoji tag sequence (ETS)*

tag_base =	U+XXXX (🔻)	New Unicode 13.0 character
tag_spec =	Q[0-9]+	Sequence of TAG characters corresponding to valid Wikidata ID, verifiable by query.

The above formulation uses the UTS #51 notation, where an underlined ASCII character stands for a corresponding Unicode character from [U+E0020 .. U+E007E] (4-bytes long in UTF-8). The default appearance indicates a missing character, and is used if the QID is not recognized or ill-formed.

Given that, here are examples of some emoji that would be valid under this additional scheme. The desired appearance would be an emoji-style image. As now, it would typically follow the "house style" for the platform in question.

Tag Sequence	Emoji
🔻Q218543♦	*White-crested tiger heron*
🔻Q459788♦	*Flag of NATO*
🔻Q673012♦	*Pisco Sour*
🔻Q20748♦	*Raclette*

Note that the vast majority of QIDs are not associated with entities that would be appropriate for emoji, such as *risk management* (Q189447) or *this* (Q3109046). As usual, it would be up to the platform to supply images via fonts. Nobody can assume that Wikidata QIDs would have associated images for the referenced entity (or that platforms would want to use such images).

Once defined in UTS #51, it would be conformant for any implementation to add an emoji like the ones above. Most implementations already handle the TAG sequences for flags (in Emoji 5.0, released on 2017-05-18), and the TAG syntax is already designed to be extensible.

Interchange

Implementations could use these tag sequences and interchange them with any other Unicode implementation. The tag sequences all would be defined as emoji, and their intended semantics are well-defined and referenceable, simply based on an examination of the tag sequence values. This is a major advantage over private-use Unicode characters or some kind of hash scheme. *Displaying them is a matter of getting a font that includes them.*

However, implementations cannot have any expectation that an *arbitrary* other implementation

would be able to render any particular QID emoji. For that to happen, the other implementation would need to have a font for that particular QID emoji.

Process

The QID emoji would not replace the current Unicode emoji encoding process, which would continue as is. Unicode would still add characters, and maintain a list of all of the Unicode RGI emoji. The one change is that we would request a QID, if available, in *new* emoji character proposals.

Where ZWJ sequences are reasonable, platforms should prefer them over the QID emoji. They are shorter and thus occupy less memory, and they have a better fallback behavior when not supported on a target implementation.

The QID emoji would offer, however, a mechanism for platforms to use for more granular or topical emoji, and take some of the pressure off the encoding of Unicode emoji characters. Platforms could support the QID emoji at any time, much as platforms can currently support emoji ZWJ sequences in whether or not they are in RGI list.

Where a QID emoji meets the other conditions of Submitting Emoji Proposals (especially the exclusions: no logos, etc), and is widely used — especially if supported by multiple major platforms — we could:

1. We could add it to the Unicode RGI list.
2. Alternatively, wide actual usage would be very strong evidence for encoding it as a regular Unicode emoji character. If we decide to provide for this, we would add an emoji data file with a mapping from defined emoji (or emoji sequences) to the corresponding original QID emoji sequence, as a convenience.

Platform Requirements

If a platform doesn't yet fully support emoji TAG sequences (est. 2017), then its support needs to be upgraded. It would also need to support the new U+XXXX character.

If an platform sees a QID emoji that it doesn't have a font for, it should display a "missing emoji" image. If any particular QID emoji is popular, the platform could add the image to its font(s), because it can determine what the QID means.

Next Steps

If this approach looks reasonable, the next step would be to create a working draft of UTS #51 that would reflect this document (as amended during UTC discussions): adding the new Tag option, making the necessary adjustments to other parts of the text and definitions, and preparing for the possible addition of U+XXXX.

That could lead up to the release in UTS #51 version 13.0, slated for in 2020Q1.

Wikidata QIDs

Wikidata IDs represent a entity of some kind, ranging from very concrete to very abstract, and point to a Wikidata "item" entry. Here are some examples of QIDs. The sample images below are just informational, and are not suitable for emoji (except for the NATO flag).

Tag Sequence	Emoji	Sample Image

Q218543	*White-crested tiger heron*	
Q459788	*Flag of NATO*	
Q673012	*Pisco Sour*	
Q20748	*Raclette*	
Q338143	*Felicia (plant genus)*	
Q234314	*Felicia Day*	
Q189447	*risk management*	*n/a*

Some features of the Wikidata QIDs are relevant to the QID emoji proposal.

- **Size/Growth.** Currently the maximum ID number is 8 digits. Wikidata's current growth rate is about 800K items per month ([dashboard](#)) so it'll eventually grow beyond that. Each extra digit in a QID has a cost in memory of 4 bytes.

- **Deprecation.** A QID can essentially be deprecated in favor of another QID by "merging" it. However, it is not removed, and remains valid.

- **Removal.** Technically removal is possible, but practically it happens very rarely, and only in cases where someone is creating a new page either for promotion or vandalism purposes, and the community decides to delete the entry.

- **Alteration.** Technically, it is possible for a Wikidata QID's page to change materially (eg, a change of the page for Q338143 changes from "*Felicia (plant genus)*" to "*Felicia Day*"). However, this is extremely rare; the community has rules against this, and they are enforced whenever discovered.

- **Timestamps.** Every MediaWiki page is versioned and every revision has an ID. That could be useful if we have any qualms about Stability of QIDs.

Issues

Length

These sequences take more memory than regular emoji (as of this writing, up to 8 digits ⇒ 42 bytes with the prefix and suffix).

We have 94 TAG values available, and could compress the decimal number into a base 64 value. For example, the 6 digits in 🦜 Q218543✦ (*White-crested tiger heron)* turn into 3 values in base 64 <2F, 16, 35>, and represented by a base emoji + 5 TAG characters — instead of a base plus 8 TAG characters. In general, the sequences would take about 30-40% less memory.

Or if length is not felt to be important, we could just use the decimal representation. However, note that the decimal notation is not particularly easier for users. The notation 1 is an abstraction representing the four-byte U+E0031, which doesn't mean anything for users (though a bit simpler in debuggers than base-64).

Tag Base

The proposal has a single tag base, a new emoji U13.0 character. An alternative approach is to allow the use of existing emoji as tag bases, such as the following:

Tag Sequence	Emoji
🦜 Q218543✦	*White-crested tiger heron*
🏴 Q459788✦	*Flag of NATO*
🥃 Q673012✦	*Pisco Sour*
🧀 Q20748✦	*Raclette*

The main problem with that approach is that there is no control on the choice of base. Take the following, for example:

🦜 Q459788✦	*Flag of NATO* emoji

It would be startling, at a minimum, for someone to see the NATO flag fallback to a 🦜. So for that case, it is far better to use as the tag_base a 🏴. But there would be no feasible way to impose constraints tag_base to make it consistent with the QID.

Moreover, there might not be an obvious choice of tag_base character: for a *falcon*, for example, an implementation might choose either eagle 🦅 or the plain bird 🦜. And others might have no obvious base, such as a *stroller*. Different platforms could choose different bases, which is clearly not good for

interoperability or consistency of fallback. On the other hand, this might sort itself out naturally, with the "first mover" effect.

Options are:

1. A new Emoji 13.0 character, as described above in the proposal.
2. An infrequently used existing symbol emoji, such as 🔤.
3. Arbitrary base characters.

Both #1 and #2 lessen the opportunity for good fallbacks, but also lessen the opportunity for misleading or inconsistent ones.

Sequences

Currently emoji tag sequences are not full-fledged emoji, in that they can't be part of other sequences. For example, they cannot appear in emoji ZWJ sequences, and thus could not be composed into longer emoji, such as in adding hair styles or gender. We could extend **ED-15a. emoji zwj element** in UTS#51 to allow them in zwj sequences and extend **ED-13. emoji modifier sequence** to allow them with skin tones.


Security

Appearance

Implementations should never put users into the situation where sensitive information depends on unique appearance of emoji in the first place.

That being said, we should place stronger requirements on the appearance of unsupported tag sequences, so that it is clear when the base character alone is present versus when the base character has TAG characters after it. See C.1.2 Sample Invalid Emoji Tag Sequences. One way of doing this is by giving U+E007F a default appearance indicating a missing-emoji glyph like the following (either as a stand-alone character or one that overlays the previous character:

That way an unsupported QID tag sequence with a MAN as a base would show up as the base followed by a missing-emoji:
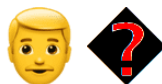
or as an overlay

Stability

Based on available information, it appears that the likelihood of unstable QIDs is sufficiently low as to not be a significant factor. If we get sufficient feedback, however, we could provide for composite tags like 🦜Q2345678-93◆, where the value after the hyphen (93) is time-stamp.

Such a time-stamp would only need coarse granularity, so it would be sufficient to be something like

Msecs since 2020-01-01 00:00:00. Thus in the example above, "-93" indicates that the representation is of the Wikidata QID Q2345678, as of the date-time **2022-12-12 09:20:00**. Having a coarse granularity means that it would be 3 centuries before we'd need more than 5 TAG characters for a time-stamp.

Thanks for feedback on this document from Jeremy Burge, Peter Edberg, Greg Welch, and Markus Scherer.