

# Mozilla Feedback on PRI #408 “QID Emoji”

Henri Sivonen, Mozilla ([hsivonen@mozilla.com](mailto:hsivonen@mozilla.com))

2020-04-15

Mozilla urges the Unicode Consortium not to adopt the QID Emoji proposal.

The proposal provides for a mechanism for minting emoji that bypasses the normal Unicode Consortium processes. We believe this would lead to problematic effects. While the foreseeable problems could be argued to have precedent in the sense that similar problems already exist in Unicode, the precedent should be viewed as problems that shouldn't be made worse and should not be viewed as a license to let the problems proliferate.

## Abstract Semantics Are Not Enough to Avoid Miscommunication

For interoperability, it's not enough to identify a concept, such as the examples given: “flag of NATO”, “gelatin dessert”, or “Triceratops”. It's also necessary to establish a representative glyph.

It's already the case with normal emoji that glyph variation is too great for users to correctly interpret the meaning that is written down in the Unicode Standard, so just having an abstract definition of meaning exist somewhere is no good if it doesn't result in glyphs that users interpret as that meaning.

For example, the name for U+1F3BF is SKI AND SKI BOOT and the reference glyph depicts a downhill ski boot and a ski in a downhill angle. Yet, Google doesn't depict the downhill ski boot but depicts skis and ski sticks that could plausibly indicate cross-country skiing, so the user-interpreted meanings of the [Google and Apple glyphs](#) are substantially different activities, so texting the emoji to someone across Android and iOS has a strong potential of miscommunication.

SKI AND SKI BOOT is by no means the only example of existing emoji whose glyph variation can lead to miscommunication. This suggests that having some repository of abstract meaning isn't enough, and instead there is a need to specify *more explicit constraints* on glyph design than what is currently specified to avoid cases like these. (In this case, the character name says SKI BOOT and Google *still* omitted the downhill ski boot.)

If the current situation of having reference glyphs and descriptive names isn't specific enough, it seems that the QID system would make things even worse.

# The Unicode Emoji Approval Process is Valuable

Compared to having clear guidance of what the glyph should look like, it is even more important to have coordination between implementors about what concepts get glyphs at all in concrete implementations.

For emoji actually to work for a communication purpose between people using different systems and fonts, it is important that the recipient has the glyphs for the characters that the sender is sending. This is already a problem when the recipient is using an Android phone whose system updates, including emoji font updates, have ended. QID emoji would likely make the situation worse.

The present emoji process is useful for ensuring that there is mutual understanding of what glyphs to extend emoji fonts with, in what time frame. That is, each new version of Unicode mints a number of emoji that vendors have the capacity to implement before the next version of Unicode.

In addition, one major benefit of emoji has actually been that it has made English-speaking developers care much more about correct Unicode support, since there are English-speakers who want to use emoji, including emoji that are outside of the Basic Multilingual Plane. The fact that new revisions of Unicode contain new emoji likewise provides pressure to keep Unicode support up-to-date, which is also valuable for having good support for *other things* added in that version of Unicode. Therefore, there is value to having an emoji process where new emoji that people care about get added in each new version of Unicode, and the emoji-driven pressure to adopt new Unicode versions leads to the adoption of the non-emoji aspects of new Unicode versions as a side effect.

## Migrations Are Costly and May Never End

The proposal suggests that popular QID emoji would get normal code point assignments later. From experience with the Web, we know that legacy data doesn't go away and we still need to deal with legacy issues that arose in the infancy of the Web. The proposal would end up with two ways to encode the same thing.

Previous migrations from PUA-mapped emoji, PUA-mapped HKSCS, and the 24 GB18030-mandated PUA mappings to non-PUA code points have been problematic. Knowingly giving rise to a continuing situation where characters are taken into use before they have normal Unicode assignments and then a migration being contemplated for later seems like something that should be avoided if at all possible.

Unlike the previous migrations, which had finite scope, the open-ended nature of QID emoji proposes a never-ending migration would probably lead to having to maintain an ever-updated new type of Unicode normalization that would replace provisional QID emoji with normal emoji over time.

# Counting Emoji in Messaging is a Privacy Problem

The notion of assigning normal code points to popular QID emoji assumes that there is a good way to gauge popularity. Emoji are very often used in private messaging, and it's not a given that public data, such as tweets, have the same emoji usage as private messages. Therefore, a scheme that relies on being able to measure which QID emoji are popular is potentially a problem in terms of the confidentiality of private messaging.

# Having to Research Existing Data is a Barrier to Entry

Standards not only provide value as a way for incumbents to negotiate new features. Standards also enable new vendors to enter the market thanks to the standards documenting what needs to be implemented in order to achieve interoperability.

When the standards don't actually fully describe what is needed in order to achieve interoperability and the would-be entrant to the market instead needs to research existing content or product behavior, this creates a barrier to entry.

Leaving it to emoji font vendors to research content out there, a substantial proportion of which in the case of emoji flows in confidential person-to-person messages, in order to figure out what set of glyphs they need to supply in order to create a competitive emoji font raises the barrier of entry compared to being able to look up a code chart at [unicode.org](http://unicode.org).

# Delegating to Vendors Should Not Increase Technical Complexity

The "Background" section of the proposal talks mainly about emoji assignment process issues, and QID emoji appears to be an attempt to apply a technical solution to a selection process problem.

Since QID emoji still need font support to work in a way that makes sense to a casual user, the likely practical outcome of QID emoji would be the delegation of emoji to the vendors whose emoji fonts have the broadest usage: Apple and Google.

Just delegating to the major vendors is bad in terms of broad-based representation in the standardization process. Even though in some sense there is nothing forcing the major vendors to adhere to the standard, so far vendors both large and small have volunteered to implement the outcome of the more broad-based standardization deliberation.

Even if the UTC wanted to delegate to the major vendors, it is unnecessary to introduce technical complexity to what kind of bits end up being interchanged. Such delegation could be achieved by the Unicode Consortium acting as a registry that hands out a code point on "just the next available number" basis from an emoji block once the major vendors have agreed to mint a new emoji.

Assigning a number should be the simplest part of the process.

Even if one agreed with delegating the decision of what gets a glyph (which appears to be a core function of the Unicode Consortium itself and, as such, odd to delegate away) to the major vendors, it seems wrong to push technical complexity onto the implementors to deal with such organizational issues.

## Flags Are Not a Role Model

One argument for QID emoji is that their technical complexity is OK, because the precedent of region subdivision flags already exists. Existing complexity is a bad reason for making the complexity proliferate.

Flags have particular political complexity that gives rise to the associated technical complexity. The mechanisms for flags avoid the Unicode Consortium having to take a stance on what the exact set of flags is. However, it seems bad to design as if all emoji required technical measures to put that kind of distance between the meaning and the UTC acknowledging the meaning considering that Unicode has been able to assign code points to ideographic characters that can be considered profane on their own without having to combine them with other characters to emerge offensive meaning as well as political symbols (such as the emblem of the Soviet Union). Therefore, the precedent suggests that the bar is rather high for having to avoid acknowledging an exact set of characters.

On the flip side, the QID emoji proposal doesn't solve issues like the gun emoji situation.

## Unlikely to Be Contained to Emoji

There doesn't appear to be a good reason to believe that if QID emoji was implemented, the mechanism would stay scoped to emoji and wouldn't be used for encoding genuinely textual characters in a way that would circumvent Unicode processes. This would result in flag emoji-like processing complexity not only an occasional emoji but potentially for almost every character in some text runs (e.g. if people started assigning a QID to every character of a given script not yet encoded in Unicode).