

**New character property: NumberofPeople**

Eduardo Marin Silva

09/04/2018

**Introduction.** Since the introduction of skin tone modifiers, people have been allowed a wider range of expression of their own identity in emoji. However, the mechanisms introduced only allowed to modify one codepoint at a time, so this meant that if an emoji consisting of a single codepoint depicted more than one person, there was no way to specify skin tones to each individual, so either there was no skin tone depicted or all persons would have the same skin tone. The consortium ruled in favor of the former, to avoid bad appearances. This proposes a new character property that allows to specify the number of people depicted so implementations can easily assign more than one skin tone modifier to a single codepoint. It is a signed integer.

**Alternatives.** Since using the ZWJ, would cause great conflicts on the current specification, I considered for a moment, encoding an invisible control character, called SKIN TONE JOINER. It would go between consecutive skin tones to denote that they apply to the same base character.

There is one main issue with this approach, and that is that implementations could easily get confused if the syntax was not exactly right, for instance if two consecutive SKIN TONE JOINER appeared, if the base character did not allow for skin tones, if there were more skin tones than people in the base character, or even if the STJ appeared in isolation. These problems are similar to the ones faced by the use of ZWJ and encoding a similar extra character, would only make the life of vendors more difficult.

**NumberofPeople.** The solution is, instead of encoding a new control character that would make the situation more complicated, we introduce a character property that consists of a signed integer.

The purpose of the property (when it is equal or greater than zero) is to indicate the backend, the maximum number of skin tones to expect to apply to the concerning character. All codepoints have a default value of zero, since most of the Unicode characters do not depict people. A value of one would apply to the grand majority of emoji depicting people (including the body parts) and a value of two would apply to characters like HANDSHAKE.

So, while normally the sequence 🧡 + 🖤 + 🧡, would display like: 🧡 🖤 🧡, or 🖤 🧡 depending of the environment, the fact that it has a value of two can be used to inform the backend to consider up to two skin tones instead of just the first one, and so, display the intended glyph (one hand dark and the other light). Greater values can be used to indicate a greater number of people, but currently most single characters, depict at most two people (an exception is FAMILY). This still allows for sequences like: 🧡 + 🖤 to still depict the intended glyph of both hands with the same skin tone and introducing any more than two skin tones would fail to render anything meaningful.

Since there is no control character to consider, it will be really easy to implement a check against mis-rendering sequences of more than one skin tone character, but all of them with the same value, such as: 🧡 + 🖤 + 🖤, in this case, it would ignore the last skin tone since it is the same and display like: 🖤 instead of 🖤 🖤 just like expected.

This has the added benefit that characters with n people, would need a maximum of n+1 characters, while the SKIN TONE JOINER implementation would require up to 2n characters.

**Negative values.** One may consider why not just making the property an unsigned integer, what would a negative number of people even mean? I choose to be a signed integer because negative values can be used to indicate edge cases. Consider the character ROCKET 🚀, all vendors design it without people, so it would naturally take a value of zero; but what if in the future, a vendor chooses to depict a person peeking through the window? Then we have a problem, because if we want to exchange information with that vendor, we would get sequences like 🚀 👤, that have distorted meaning to everybody except that vendor; after all, rockets don't have skin like humans.

In that case we can assign a value of -1 to the property, which in this context literally means: "usually depicted with 0 people but may be depicted with up to one person". Vendors will be able to avoid displaying the skin tone modifier, without fearing loss of information, so the sequence: 🚀 + 👤 will displaying like: 🚀 except for the vendors that choose to show a human (since the skin tone is just hidden from view and not deleted, it allows for full roundtrip compatibility).

The negative value can also be used for emojis that undeniably show a human like character: like the ZOMBIE 👤, but the use of skin tones is optional. In the case of the zombies, only Microsoft has added support for skin tones: 👤, but other vendors may see that as unnecessary, and so will like to exchange information with Microsoft applications, without having to display an ugly skin tone character beside them. The property indicates when it is safe to do that.

Lower values can be used to denote the possibility of expecting that amount of skin tone characters.

The third use for it would be cases where it is truly debatable whenever some character intends to represent a person or not (like the BRAIN 🧠); for those, the value can be set to -1 or lower.

**Stability.** The property can be changed between versions of the standard, so in our rocket example, if the majority of vendors decided to display a person, following the lead of our fictional vendor, then it would be safe to change the value from -1 to 1. For BREASTFEEDING 👤 an initial implementation may set the value to 1, since that is the maximum number of skin tones used usually, but can always be changed to 2, if there is a need to specify the skin tone of the baby, separately (although I would suggest setting it to 2 initially).

**Eligible characters.** I am expanding the property beyond emoji use, to all pictograms depicting people. The reason why I am not restricting it to emoji, is because a character property applies to all characters and we want its semantic to remain intact.

It may also be somewhat helpful, for instance, in counting the amount of people in a typical Egyptian hieroglyph string or highlighting only the human body parts in a signwriting layout.

With respect to skin tones, all the inclusion of these non-emoji characters does, is to make vendors also expect, characters depicting people with zero skin tones (something that is already the case for emoji like the MOTORCYCLE 🏍️). This is consistent with our rule of being less or equal to.

**Candidates for value 1.** The majority of people emoji fit this category, even when not the whole person is shown such as EAR 👂, but it also includes some non-obvious cases, like WHEELCHAIR SYMBOL ♿, PLACE OF WORSHIP 🕌, grand part of Signwriting characters, and many Egyptian and Anatolian hieroglyphs.

All smileys can either be 1, 0 or -1, depending on the consortium's preferences.

**Candidates for value 2.** This can include the following emojis:

- HANDSHAKE 🤝
- WRESTLERS 🤼
- WOMAN WITH BUNNY EARS 🐰
- FACE MASSAGE 🤲
- BREASTFEEDING 🤱
- MAN AND WOMAN HOLDING HANDS 🤝
- TWO MEN HOLDING HANDS 🤝
- TWO WOMEN HOLDING HANDS 🤝
- COUPLE WITH HEART 💑

It also includes some symbols like RESTROOM 🚻. It is not out of the question that some hieroglyphic systems may have characters with more than one person.

**Candidates for value 4.** The only candidate here is the character FAMILY 🧑👦, I choose 4 instead of 3 based on things like the code chart glyph and the alternative name "Family with Boy and Girl", our implementation can then be more inclusive, since we can always specify a number of skin tones lower than 4 if the glyph demands it.

**Candidates for value 5.** Here the only candidate is MODERN PENTATHLON 🏅, since its glyph may depict up to 5 persons.

**Candidates for value -1.** Here we include different characters for different reasons.

Body parts - Because they either they may be represented with surrounding skin or there is a debate as to wherever it truly represents a person:

- FOOTPRINTS 👣
- EYES 👁️
- EYE 👁️
- BRAIN 🧠
- MOUTH 🗨️
- OBSERVER EYE SYMBOL 👁️

These fantasy beings: Because the non-realistic skin tones are merely optional:

- VAMPIRE 🧛
- ZOMBIE 🧟
- GENIE 🧞

**Further expansions.** The property may be expanded to cover ideographs meant to represent a person, or even script specific symbols, however that is beyond the scope of this work and I see that as problematic. For now, only pictograms shall be considered, not abstract symbols.