L2/

Re: **Appearance of unsupported emoji sequences**
From: **Mark Davis, ESC**
Date: **2017-03-21**

Summary

The set of supported emoji sequences may vary by platform. We would improve communication if in a future version of UTS #51 we encouraged implementations to clearly indicate unsupported sequences, especially for emoji ZWJ sequences such as a *ninja cat* or *pirate flag*.

Background

For example, take the following emoji zwj sequence.

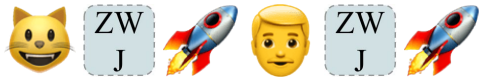
On a particular platform, it can be shown as something the following:

However, if that fails (most platforms), it shows up as:

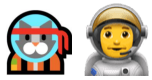
Now, it is by long-standing Unicode design that ZWJ sequences show up as their components if the platform doesn't support them. It does give the user an indication of what the meaning is.

However, as normally implemented it is indistinguishable from the sequence *without* the ZWJ. That is, the user sees no indication that this is anything but the plain sequence of characters (that is, without ZWJ).

Similarly, someone on an iPhone or Android has no indication that the following:



Was intended to be seen as two emoji, <*ninja cat*><*astronaut*>:



Not as the three emoji <*cat*><*rocket*><*astronaut*> (which is what you see on a recent iPhone or Android):

Proposal

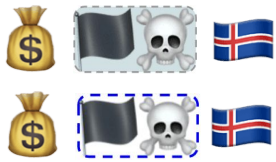
The proposal is to add language to UTS #51 recommending that an indication of the combined nature of an unsupported sequence be used *where possible*. There are a number of advantages to doing so:

1. The user would know that the original intention was a sequence: that a sequence was intended not as 4 images for <money><flag><skull><iceland>, but as 3 images for <money><flag + skull><iceland>.

2. The user would realize why the behavior is different:
    a. Why the arrow key won't move between the flag and the skull & crossbones, or between the cat and the rocket
    b. Why the sequence doesn't line-wrap between apparently separate emoji.

There are a few basic approaches that could be taken by the rendering engine to do this that we could illustrate in the text using the following. We should also solicit feedback from rendering system designers on the feasibility of different approaches and from graphic designers for what would be clearest for users.

1. Cartouche — the components are rendered as if there were no ZWJ (the fallback representation), but a "cartouche" style is provided for them as a background, by the rendering engine. The metrics would be the same as the sequence of components; they would just have a backdrop.



The cartouche background could also be applied to the stacked form, or used for other invalid emoji sequences, such as an invalid emoji tag sequence, or invalid emoji modifier sequence.

2. Stacking — a single, artificial emoji is produced by the rendering engine, tiling the components into a square with a bit of overlap (suggested ordering: clockwise from northeast). The metrics would be the same as the desired outcome. Because the emoji are stacked, they might be as easy to see at small sizes as the Cartouche form.

3. Both — Same metrics as Stacking, but with a Cartouche visual indication of "fallback" nature.