Noto Color Emoji, and the sequences it draws

Examples 2.14 and 2.17 build the colour face they show, because a page about a file format reads better with the file on it. This one takes a face off the shelf instead: Noto Color Emoji, 25 MB of artwork, 3993 colour glyphs of COLR version 1, and the face nearly every colour font measurement in this package was taken against.

An emoji face does not draw one picture per character. It draws one picture per SEQUENCE, and which characters make one up is a property of the FACE rather than of Unicode. So -chars takes text: the string goes to the face whole, the "ccmp" feature of the face says where the joins are, and every glyph that comes back is one glyph of the Type 3 font - with one advance width, and with the whole sequence in its /ToUnicode.

The row example 02.09 sets in Noto Emoji, the MONOCHROME face of the same family, set here in the colour one - the same twenty-one marks, character for character. Twenty-three code points make twenty-one pictures: the desert island carries U+FE0F and the chilli U+FE0E, the text presentation selector, and this face has a glyph for neither - each rides along with the mark in front of it and comes back out of the /ToUnicode.

One character each - and the last one is two, because a heart carries the emoji presentation selector U+FE0F behind it.

The same seven at twice the size. Every mark on these pages is a PATH - a Type 3 glyph description of clipped shapes and shadings, not a picture - so it scales without losing anything, and the only difference between this row and the one above it is the text matrix. Measured off this document: 7.906 mm at 18 pt against 15.813 mm at 36 pt, which is exactly twice. [pdfimages -list] on the finished file lists nothing at all.

Two to seven characters each: two families, two flags of regional indicator pairs, the transgender flag with its joiner and its two selectors, and the Scottish flag - a black flag and six invisible tag characters spelling "gbsct". The skin tones have a page of their own overleaf.

One sequence, one glyph, one width

The measurement below is the whole claim of this page, and it is read off the document rather than asserted: [textWidth] answers the same number for a sequence of seven characters as for a single emoji, because in the font they are the same thing - one glyph, one entry in the /Widths array, one byte in the string that sets it. A font that had taken the sequence apart would measure seven times as wide and draw seven pictures.

mark	characters	width	sequence
😀	1	4.392 mm	U+1F600
❤️	2	4.392 mm	U+2764 U+FE0F
🇩🇪	2	4.392 mm	U+1F1E9 U+1F1EA
👍🏽	2	4.392 mm	U+1F44D U+1F3FD
👨‍🌾	4	4.392 mm	U+1F9D1 U+1F3FD U+200D U+1F33E
👩‍🦰	5	4.392 mm	U+1F477 U+1F3FD U+200D U+2640 U+FE0F
👨‍👩‍👧	5	4.392 mm	U+1F468 U+200D U+1F469 U+200D U+1F467
🏳️‍⚧️	5	4.392 mm	U+1F3F3 U+FE0F U+200D U+26A7 U+FE0F
👨‍👩‍👧‍👦	7	4.392 mm	U+1F468 U+200D U+1F469 U+200D U+1F467 U+200D U+1F466
🏴	7	4.392 mm	U+1F3F4 U+E0067 U+E0062 U+E0073 U+E0063 U+E0074 U+E007F

Ten units, one to seven characters each, and one width for all of them: 4.392 mm at 10 pt. The last row is seven characters wide and one glyph big, and the two in the middle carry a skin tone and a joiner at once.

The skin tones, one icon at a time

A skin tone is not a picture. U+1F3FB to U+1F3FF are the five Fitzpatrick MODIFIERS, and a modifier changes the mark in front of it: the base emoji and the modifier are one sequence, one glyph in the face and one glyph here, with one advance and one /ToUnicode entry carrying both code points back. The three rows below are the same icon through the whole ramp, so that what the modifier does is visible rather than described.

The base on its own, then the same base through all five modifiers: U+1F44D, and U+1F44D followed by U+1F3FB, U+1F3FC, U+1F3FD, U+1F3FE, U+1F3FF. Six units, eleven code points, six glyphs - every one of them the same width, because a modifier adds nothing to the advance.

The same ramp on a ZWJ SEQUENCE, which is the case that matters: the builder is U+1F477, a zero width joiner, the female sign U+2640 and a presentation selector, and the modifier goes in as the SECOND code point rather than at the end. Six units - four code points for the first and five for the other five - six glyphs, one width. A match that took the base and the tone and then stopped would draw a toned builder and then refuse the joiner; one that matched neither would draw five pictures where the face draws one.

And once more with a shorter sequence around the tone: a person, a sheaf of rice and the joiner between them. Six units of three and four code points, six glyphs, one width. Every unit on this page was checked against hb-shape - the glyph this package picks for a sequence is the glyph HarfBuzz picks.

What this face paints with

Measured over all 3993 colour glyphs of this face: 129823 clipped shapes, 111668 solid fills, 35981 transformations, 9844 radial gradients, 8627 linear ones, 578 composites - and NOT ONE SWEEP. The angular gradient that costs example 2.17 a fan of Gouraud triangles, because PDF has no shading type for it, does not occur in this face at all. The composites are of two modes only, and both of them sit in a flag: Source In, which pours a gradient into the shape of something else, and Soft Light, which is a PDF blend mode under another name.

- 🎯 layers of solid fills and nothing else - the common case, by a wide margin
- 😀 a radial gradient in the face, and PDF's type 3 shading on the way out
- 🌍 four radial gradients under four transformations
- 🔮 five linear gradients and two radial ones in one glyph
- 🇩🇪 a linear gradient, a Source In and a Soft Light - both of this face's composite modes in two characters
- 🏴 the same three, reached through six invisible tag characters

In the running text

One string, one call, two faces: -fallback names the faces that may set what -family cannot, and the chain offers a SEQUENCE as one unit - longest match first - rather than a character at a time. That is what lets a family emoji stand in the middle of a sentence instead of breaking apart at its joiners, and it is why the marks below need no space around them: a sequence glued to the word in front of it is still one glyph.

The crate📦 left the depot🚚 on Tuesday and reached the family👨‍👩‍👧 in 🇩🇪 on Thursday, who signed for it👍🏽 at 14:20✅. Two pallets went back⚠️ - the 🇫🇷 consignment is still on the water, and the 🏴 one has not left the yard.

The characters that draw nothing

Of the 1499 characters this face's cmap maps, 39 draw nothing at all: the space, the zero width joiner, and the 37 tag characters a flag like the Scottish one is spelled with. They are real entries in the cmap and they have real glyph numbers, but their outlines are empty and the COLR table says nothing about them - so a font built from one would be valid, extractable and blank, with nothing reporting it. That is refused by name, and only where such a character is asked for ALONE: inside a sequence the joiner and the tags are swallowed by the ligature and come back out through the /ToUnicode, which is what the last row of the width table on the first page shows.

The message, caught from this very page: tclpdf: character U+200D is glyph 4005 of the face, the "COLR" table draws no colour layers for it, and its own outline is empty - so the glyph would draw nothing, and a font built from it comes out blank with nothing reporting it. Most glyphs of a colour font are layers or plain outlines; this one is neither

Check it yourself

The first command hands the sequences back unchanged - joiners, skin tones, selectors and tag characters included - which is the /ToUnicode of a Type 3 font doing its work. The second lists the pictures in the file and prints a header with nothing under it: every emoji on these pages is a path in a glyph stream, and there is no bitmap anywhere in the document. The third is the font's own table of glyph procedures, and its keys spell the sequences out one code point at a time - one procedure for all seven characters of the Scottish flag. The fourth counts the shadings the gradients became.

```
pdftotext 02.18-noto-color-emoji.pdf - | head -20
pdfimages -list 02.18-noto-color-emoji.pdf
qpdf --qdf --object-streams=disable 02.18-noto-color-emoji.pdf - | grep -a -A22 /CharProcs
qpdf --qdf --object-streams=disable 02.18-noto-color-emoji.pdf - | grep -ac ShadingType
```