

Reading order

Reordering is not shaping, and needs no shaper.

PDF has no notion of direction: a show-text operator paints glyphs at the pen and moves the pen along. So the library reorders before it draws, and a caller who takes no HarfBuzz dependency still gets Hebrew and Arabic the right way round - unjoined, which is wrong, but no longer also backwards, which was the older complaint.

Hebrew, drawn by the sans face:

שלום

The same letters, one call each, left to right:

ש ל ו ם

Which is the whole of it.

Read those two lines against each other: the word is the row of grey letters in reverse. The first letter written is the rightmost one drawn, and nothing in the calling code said so - the string went to DrawString in the order it is typed.

A Hebrew word inside an English sentence:

The word שלום means peace.

An English word inside a Hebrew sentence:

שלום peace שלום

Guessed from the text, then declared right to left:

שלום ACME

שלום ACME

The name moves, and the letters do not.

The first line opens with a Latin word, so the paragraph is guessed left to right and the name is laid down first, at the left. Declaring it right to left lays the same first word down at the right-hand end instead, where a Hebrew reader begins, and the Hebrew after it - reading leftwards. Neither line reverses its own characters; what changes is which end the line is built from, which is what the paragraph level says.

Shaping

A shaper is registered, so these letters join.

An Arabic letter takes a different form according to what it sits between - initial, medial, final or isolated - and which form is which is a rule inside the font, in its GSUB table. Reading that table is what a shaper does. Install PdfPinata.HarfBuzz and set `GlobalFontSettings.TextShaper` and this page draws joined letters; leave it unset and the same string comes out as isolated forms, correctly ordered.

The name of the script, in it:

عربية

A greeting:

مرحبا

The same five letters, asked not to join:

مرحبا

U+200C, the zero-width non-joiner.

It is bidirectional class BN, so rule X9 removes it before the algorithm resolves anything - and it still has to reach the shaper, because it is the character that says how the letters on either side of it join. So it is inside the run and not on the page: no glyph is drawn for it, and the two rows differ only in shape.

What the pieces are:

TextItemizer cuts a string into runs of one direction and one script. ITextShaper is handed one run at a time, with its script and its language. ShapedGlyph.Cluster maps each glyph back to the characters it stands for, which is what writes `/ToUnicode` - so the text can still be selected and copied out of the finished page, even where one glyph swallowed two characters.

Fallback

A fallback is registered.

Every line on this page asks for the sans face, which has not one Arabic glyph. Without a fallback each Arabic character is .notdef - an empty box - with no warning at any layer, even though a face that could draw it is loaded in the same process. GlobalFontSettings.FontFallback says which families to try instead.

A Latin face asked to draw Arabic:

مرحبا

Mixed, in one string and one DrawString call:

Peace, مرحبا, shalom.

Two faces on one line, and both embedded in the file.

The run is cut where coverage changes, each piece is drawn by the face that has the character, and the face the caller asked for is selected again at the end. A space is never given a face of its own: it carries no shape worth choosing one for, and claiming it would split a sentence into one run per word and lose the shaping across every one of the boundaries.

XTextFormatter, declared right to left and justified:

שלום مرحبا שלום مرحبا שלום مرحبا שלום مرحبا שלום
مرحبا

Justifying is the alignment that needed changing.

Every other one hands a whole line to DrawString, which orders it. Justifying places each word itself, so it has to ask where each belongs - and order them by the leftmost position any of their characters ends up at rather than by the first, because a right-to-left word's first character is its rightmost one.