

What each encoding carries

An XFont carries an XPdfFontOptions, and the encoding in it decides what kind of font object is written into the PDF. WinAnsi writes a simple font with a single-byte encoding and 256 places to put a character in; Unicode writes a CID font, which has no such limit. Both rows below are the same string in the same face at the same size. Where they differ, the character was not one of WinAnsi's 256.

WinAnsiDefault

The quick brown fox

Plain ASCII - both encodings carry it

Café, naïve, Straße, £42

Latin-1 - inside WinAnsi's 256 places

Greek - outside WinAnsi

Cyrillic - outside WinAnsi

wiczenia, ód

Latin Extended - outside WinAnsi

Arrows and mathematics - outside WinAnsi

UnicodeDefault

The quick brown fox

Café, naïve, Straße, £42

Ελληνικά

Кириллица

Ćwiczenia, Łódź

→ ← ↑ ↓ ∑ ∞

Nothing threw. A character WinAnsi has no place for is not an error - it is dropped or replaced on the way out, which is exactly the failure that gets noticed after the document has been sent. Unicode is the safe default and is what PdfDocumentRenderer(unicode: true) selects for a PinataLayout document, under a name that gives no hint that it is this setting.

What it does to the file

Five one-word documents, saved and reopened, with the font object each of them wrote read back out. The subtype is the kind of PDF font; the key is where the face's own bytes ended up.

document	subtype	key	face bytes	file
Liberation Sans, WinAnsi	/TrueType	/FontFile2	8,959	11,672
Liberation Sans, Unicode	/Type0	/FontFile2	7,835	10,118
Liberation Sans, Unicode, Cyrillic	/Type0	/FontFile2	8,225	10,581
Source Code Pro, WinAnsi	/TrueType	/FontFile3	90,338	93,039
Source Code Pro, Unicode	/Type0	/FontFile3	90,338	92,609

Two outlines, two embedding paths

Liberation Sans is TrueType, so only the glyphs the document uses are embedded - the face is subsetting, and the two Liberation rows differ in size for that reason alone. Source Code Pro has PostScript (CFF) outlines, which this library embeds whole under /FontFile3 because a CFF subsetter is not written; a document using one character of it carries the same bytes as a document using all of them. That is the trade recorded in font-embedding-gaps.md, and it is why the size column moves for one face and not for the other.

Why CJK is not on this page

Nothing here is a limit of the library: a CID font carries any character a face has a glyph for, and this app simply does not carry a face that has CJK glyphs. Liberation Sans covers Latin, Latin Extended, Greek and Cyrillic, which is what page one uses. A CJK face is several megabytes for one panel, so the demo shows the mechanism and leaves the asset to whoever needs it - register any face through IFontResolver and the Unicode path above carries it.