

I just found (and fixed) a bug in mupdf.

Routine `loadcolorkey` ("fitz\mupdf\pdf_image.c" line 181) and defines `(tbit, ttwo, tfour` in "fitz\raster\imageunpack.c" line 93) scale palette pixel values.

Thus a 4 bit/pixel bitmap pixel with value 14 becomes $14 * 17 = 238$.

Routine `pdf_loadtile` ("fitz\mupdf\pdf_image.c" line 514) tries to undo the scaling by multiplying the scaled pixel value with $65536/scale$.

Thus the pixel palette index from above becomes `HIWORD((65536/17)*238) = 13`. That completely mixes up colors.

I understand that multiplying is faster than dividing, but a scale factor of 16 for 4 Bit, 64 for 2 Bit and 128 for one bit would help better.

I don't know for what the scale is good for anyway. Leaving the pixels untouched would be speedy and correct.

Comment 1 by [kkowalczyk](#). Today (5 minutes ago)

Could you post a PDF that shows this problem somewhere? That would help convince mupdf people that the fix is right.