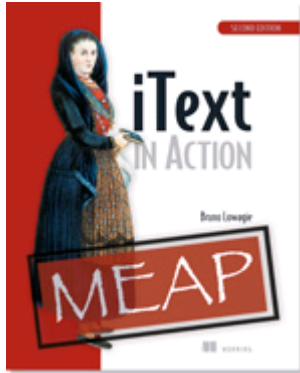


Replacing a font

Article based on



[iText in Action, Second Edition](#) EARLY ACCESS EDITION

Bruno Lowagie

MEAP Release: November 2009

Softbound print: Fall 2010 | 600 pages

ISBN: 9781935182610

This article is taken from the book iText in Action, Second Edition. The author explains how to replace a font embedding a font file after the fact.

Tweet this button! (instructions [here](#))

Get **35% off** any version of [iText in Action, Second Edition](#) with the checkout code **fcc35**. Offer is only valid through www.manning.com.

Figure 1 shows two PDF files that were created in exactly the same way, except for one difference: in the top PDF, the font (Walt Disney Script v4.1) wasn't embedded. It's a font I've downloaded from a site with plenty of free fonts. The font isn't installed on my operating system. As a result, Adobe Reader doesn't find it, and the words "iText in Action" are shown in Adobe Sans MM, which is quite different from the font shown in the PDF that has the font embedded.



Figure 1 Non-embedded versus embedded font

If we had the top PDF as well as the font file for the Walt Disney Script font, we could use listing 1 to embed that font after the fact.

Listing 1 EmbedFontPostFacto.java

```

RandomAccessFile raf = new RandomAccessFile(FONT, "r");           A
byte fontfile[] = new byte[(int)raf.length()];                 A
raf.readFully(fontfile);                                       A
raf.close();                                                    A
PdfStream stream = new PdfStream(fontfile);                     B
stream.flateCompress();                                         B
stream.put(PdfName.LENGTH1, new PdfNumber(fontfile.length));   B
PdfReader reader = new PdfReader(RESULT1);
int n = reader.getXrefSize();
PdfObject object;
PdfDictionary font;
PdfStamper stamper
    = new PdfStamper(reader, new FileOutputStream(RESULT2));
PdfName fontname = new PdfName(FONTNAME);
for (int i = 0; i < n; i++) {                                   C
    object = reader.getPdfObject(i);                             C
    if (object == null || !object.isDictionary())               C
        continue;                                              C
    font = (PdfDictionary)object;                                C
    if (PdfName.FONTDESCRIPTOR.equals(font.get(PdfName.TYPE))   C
        && fontname.equals(font.get(PdfName.FONTNAME))) {        C
        PdfIndirectObject objref = stamper.getWriter().addBody(stream); D
        font.put(PdfName.FONTFILE2, objref.getIndirectReference()); E
    }
}
stamper.close();

```

- A Reads the font file into a byte array**
- B Creates a PDF stream**
- C Finds the unembedded font**
- D Adds the stream to the writer**
- E Adds the reference to the stream**

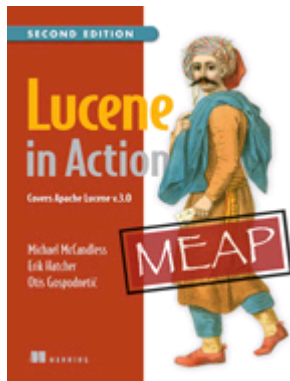
Note that we're adding the complete font file in this example. We add the reference to the stream using the key `FONTFILE2` because we know in advance that the font has True Type outlines. That's not the only assumption we make. We also assume that the metrics of the font that is used in the PDF correspond with the metrics of the new font we're embedding.

As for parsing PDF, we could only do a fair attempt, but the functionality could fail for PDFs using exotic encodings. In real world examples, replacing one font with another can be very difficult.

Summary

In this article, I showed you how to replace a font in a PDF document using the embed feature in iText.

Here are some other Manning titles you might be interested in:



[Lucene in Action, Second Edition](#)

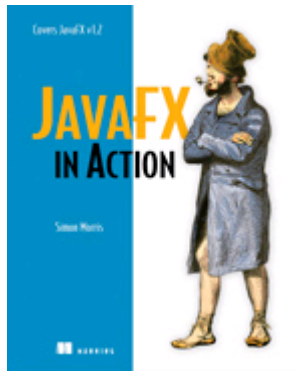
EARLY ACCESS EDITION

Bear P. Cahill

MEAP Began: May 2008

Softbound print: May 2010 | 475 pages

ISBN: 1933988177



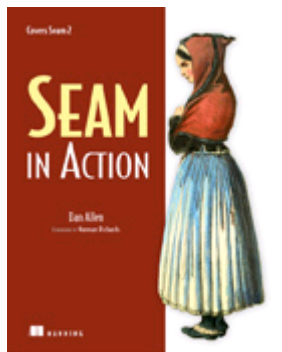
[JavaFX in Action](#)

IN PRINT

Simon Morris

Softbound print: November 2009 | 384 pages

ISBN: 1933988991



[Seam in Action](#)

IN PRINT

Dan Allen

Softbound print: September 2008 | 624 pages

ISBN: 1933988401