PDFBOX-5018

m.barbi@satanet.it

2020-11-17

Contents

Chapter 1

Root issue PDFBOX-5018

1.1 Summary

Wrong extraction of blank character

1.2 Description

Applying the PDFTextStripper to the attached PDF Document, a not-existing blank character is read in the following text:

!image-2020-11-17-11-19-45-132.png!

Instead of "O1AI7A" the text supplied to the writeString method callback is "O 1AI7A".

Making copy&paste from Adobe Reader doesn't introduce any blank character.

1.3 Attachments

1. image-2020-11-17-11-19-45-132.png
2. IT00820340966_Z-SO-PO 5270213 (1).pdf

1.4 Comments

1. **mkl:** The instruction drawing that code is:

```

[( )250(O1AI7A)] TJ
```

I.e. first a space is drawn, then the cursor is moved left by 250 units, then the actual code is drawn.

So first of all this is not *a not-existing blank character* we're looking at.

Furthermore, the width of the space in the font at hand is exactly 250 units, too. Thus, essentially the {{O}} and the space start at the same position. In the extracted string, though, two characters cannot share a position, so one of {{O}} and the space must come first. In your case the {{O}} won.

2. **m.barbi@satanet.it:** Thanks for the detailed explanaition.

   Then, at our end we could implement a logic that, if two characters share the same location and one of them is a blank, the blank is always rendered as first. This is not a general rule but in our use cases may be effective.

3. **tilman:** The effect is only when sorting. Seems the order is undefined when the positions are identical.