

Remove the margins in the header using Apache POI in Java
Asked 2 years, 7 months agoModified 2 years, 4 months agoViewed 945 times

I am using Apache POI version 4.1.2. When, I try to create a document with an image in the header, there is some default margins allocated for the header on the 4 sides of the image.

How can I remove the default margins in the header portion and place the image fully occupied in the header portion?

```
XWPFDocument document = new XWPFDocument();
CTSectPr sectPr = document.getDocument().getBody().addNewSectPr();
XWPFFooterPolicy headerFooterPolicy = new XWPFFooterPolicy(document, sectPr);
XWPFFooter header = document.createHeader(HeaderFooterType.FIRST);
XWPFPParagraph paragraph = header.createParagraph();
XWPFRun run = paragraph.createRun();
```

//Adding the image using run.addPicture() method.

The output document is looking like below,



I would like to remove the margins and make that image fully occupied in the header.

Edit on 27th July, 2022:

I have specific header for the first page and no headers for all other pages. For creating that Header in my document, I use

```
XWPFFooter header = document.createHeader(HeaderFooterType.FIRST);
```

Also, I have some image in footer which should be applied for all the pages, including the first page. For that, I use

```
XWPFFooter footer = document.createFooter(HeaderFooterType.DEFAULT);
```

But, the footer is not getting applied to the first page alone and gets applied to all the other pages. I have gone through the Apache documentation of the constant HeaderFooterType.DEFAULT and it says,

This is the default header or footer, It is displayed on every page where a more specific header or footer is not specified.

So, I created two footers like below, one is for First page and rest is for all the other pages to add the same footer content in all the pages.

```
XWPFFooter footer = document.createFooter(HeaderFooterType.FIRST);
XWPFFooter footer1 = document.createFooter(HeaderFooterType.DEFAULT);
```

Is there any workaround for this? Or am I missing something to make it work with single footer instance with HeaderFooterType.DEFAULT. Please advise. Thanks.
[javaapache-poi](#)

ShareImprove this questionFollow

edited Jul 27, 2022 at 14:47

asked May 18, 2022 at 11:06

Heyyou
311 1 4 16

3 Please read [How to create a Minimal, Reproducible Example](#). To be able to help we need the code which leads to the described problem. i.e. the code where you try to create the document having an image in the header. – Axel Richter May 18, 2022 at 12:10

Axel Richter, I believe the question is very well described. Just, trying to place a image in the header portion of the word document using apache POI. In that situation, I would like to neglect the white space in the header portion to make the image fully occupied in the header portion. Hope this comment helps. – Heyyou May 18, 2022 at 13:09

2 With no code, it's very hard to see what you're doing wrong... Without the code that adds the picture, it's tricky to see what needs to be changed – Gagravarr May 18, 2022 at 13:58

Add a comment

1 Answer

Sorted by:

Highest score (default)

Using XWPFRun.addPicture the image is inline with text so the paragraph settings matters. Word paragraphs have spacing after it per default.

Using XWPFPParagraph.setSpacingAfter it can be set 0. Also Word paragraphs have spacing between lines in paragraph per default.

Using XWPFPParagraph.setSpacingBetween it can be set to single, so there is no spacing between the lines in paragraph.

To make the picture in the header appear at absolute top of the page, the top page margin needs to be 0. Also the distance between header and page margin needs to be 0. Unfortunately setting the page size and page margins is not yet implemented in XWPF. So we need using the org.openxmlformats.schemas.wordprocessingml.x2006.main.* classes.

The gaps at left and right of the picture are the distances between the picture's left edge and the left page margin respective the picture's right edge and the right page margin. It depends on the picture's width. But I cannot see how this gaps shall be removed without violating the picture's aspect ratio and so distort the picture. If that is the wanted, then set the picture's width appropriate.

To make the picture full width, one need to know the needed width in points (pt). That would be the inner width of the page between the page margins. In my example 8.5" page width - 0.5" left margin - 0.5" right margin = 7.5".

Of course, there is a left page margin and a right page margin too, which **never** should be set 0. That would make the document not printable on most of the printers as printers have non printable ranges at left. There are non printable ranges at top, right and bottom as well, but this is easier to handle as the non printable range at left. To shift content to right from left page margin, each single content row would must be shifted.

Because of set the top margin 0, that affects all pages. So one would need to put something in the headers of the other pages to shift the body content down. That could be some content or an empty paragraph having a spacing after.

Complete example:

```
import java.io.FileOutputStream;
import java.io.FileInputStream;
import java.io.IOException;

import org.apache.poi.util.Units;

import org.apache.poi.xwpf.usermodel.*;

import org.apache.poi.xwpf.model.XWPFHeaderFooterPolicy;
import org.apache.poi.wp.usermodel.HeaderFooterType;

import org.openxmlformats.schemas.wordprocessingml.x2006.main.CTSectPr;
import org.openxmlformats.schemas.wordprocessingml.x2006.main.CTPageSz;
import org.openxmlformats.schemas.wordprocessingml.x2006.main.CTPageMar;

import java.math.BigInteger;

public class CreateWordHeaderFooter {

    public static void main(String[] args) throws Exception {

        XWPFDocument document = new XWPFDocument();

        // the body content - two pages
        XWPFParagraph paragraph = document.createParagraph();
        XWPFRun run=paragraph.createRun();
        run.setText("The Body:");

        paragraph = document.createParagraph();
        run=paragraph.createRun();
        run.setText("First Page....");

        paragraph = document.createParagraph();
        run=paragraph.createRun();
        run.addBreak(BreakType.PAGE);

        paragraph = document.createParagraph();
        run=paragraph.createRun();
        run.setText("Second Page....");

        // first page header
        XWPFFooter header = document.createHeader(HeaderFooterType.FIRST);

        paragraph = header.getParagraphArray(0);
        if (paragraph == null) paragraph = header.createParagraph();
        paragraph.setAlignment(ParagraphAlignment.CENTER);

        // set spacing after paragraph 0
        paragraph.setSpacingAfter(0);
        // set spacing between lines in paragraph to 1 (single)
        paragraph.setSpacingBetween(1d, LineSpacingRule.AUTO);

        // the image is inline with text so the paragraph settings matters
        run = paragraph.createRun();
        String imgFile="./laptop.jpg";

        // calculate page inner width to set the picture's width the same
        long pageInnnerWidthPt = Math.round(7.5 * 72d); //8.5" page width - 0.5" left margin - 0.5" right margin = 7.5"
        run.addPicture(new FileInputStream(imgFile), XWPFDocument.PICTURE_TYPE_JPEG, imgFile, Units.toEMU(pageInnnerWidthPt), Units.toEMU(200));

        // default page header
        header = document.createHeader(HeaderFooterType.DEFAULT);
        paragraph = header.getParagraphArray(0);
        if (paragraph == null) paragraph = header.createParagraph();
        // set spacing after to 24 pt to shift the body down in default pages
        paragraph.setSpacingAfter(24*20);

        // create footer start
        XWPFFooter footer = document.createFooter(HeaderFooterType.DEFAULT);

        paragraph = footer.createParagraph();
        paragraph.setAlignment(ParagraphAlignment.CENTER);

        run = paragraph.createRun();
        run.setText("The Footer:");

        // create page margins
        CTSectPr sectPr = document.getDocument().getBody().getSectPr();
        if (sectPr == null) sectPr = document.getDocument().getBody().addNewSectPr();
        CTPageSz pageSz = sectPr.addNewPgSz(); // paper format letter
        pageSz.setW(BigInteger.valueOf(12240)); //12240 Twips = 12240/20 = 612 pt = 612/72 = 8.5"
        pageSz.setH(BigInteger.valueOf(15840)); //15840 Twips = 15840/20 = 792 pt = 792/72 = 11"
```

```

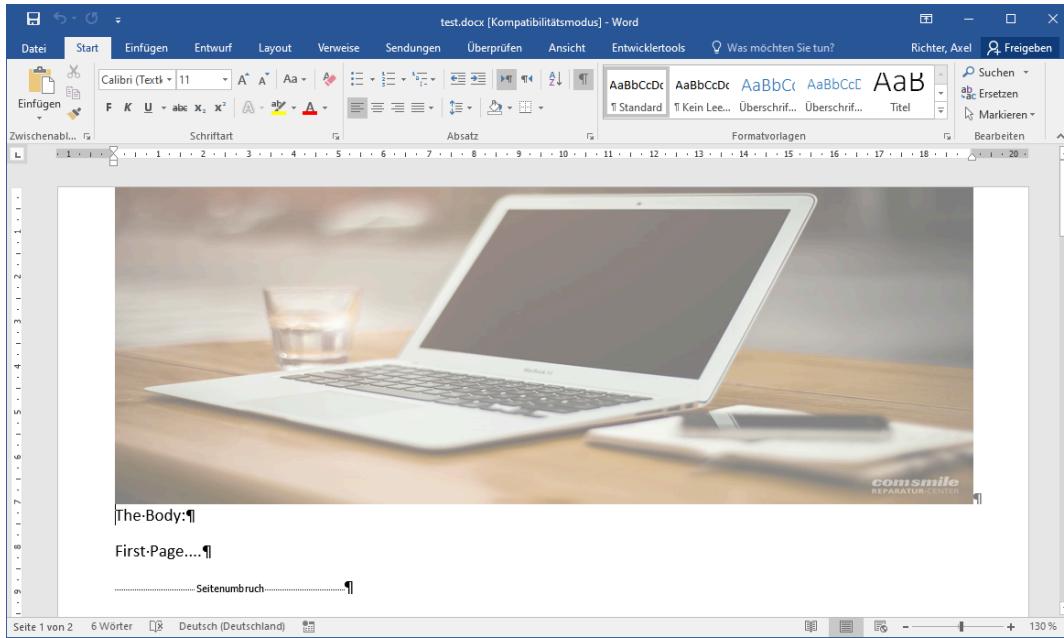
CTPageMar pageMar = sectPr.getPgMar();
if (pageMar == null) pageMar = sectPr.addNewPgMar();
pageMar.setLeft(BigInteger.valueOf(720)); //720 TWentieths of an Inch Point (Twips) = 720/20 = 36 pt = 36/72 = 0.5"
pageMar.setRight(BigInteger.valueOf(720));
// set top page margin 0, so header can be at absolute top
pageMar.setTop(BigInteger.valueOf(0));
//pageMar.setBottom(BigInteger.valueOf(0));
//pageMar.setFooter(BigInteger.valueOf(0));
// set distance between header and page margin 0, so header starts at absolute top
pageMar.setHeader(BigInteger.valueOf(0));
//pageMar.setGutter(BigInteger.valueOf(0));

FileOutputStream out = new FileOutputStream("./test.docx");
document.write(out);
out.close();
document.close();

}
}

```

Result:



Only left gap is the left page margin, which should not be touched. And only right gap is the right page margin, which also should not be touched.

ShareImprove this answerFollow

edited Jul 25, 2022 at 13:41

answered May 18, 2022 at 15:37

 Axel Richter
61.7k 7 67 93

Above solution does not fix the left margin issue even though I increase the width of the image (But the white space in the right side of the image is filled with image when the width is increased). Left side white space is still there. Also, if I set pageMar.setTop to 0, the contents in the second, third page(which does not have any headers) is also getting displayed from the absolute top as like the image is displayed in your screenshot. Any possible solution for that? – Heyyou Jul 25, 2022 at 11:41

@Kishore: How would you fulfill your requirements using Microsoft Word GUI? Apache POI cannot create a Word document which Microsoft Word is not able to create. – Axel Richter Jul 25, 2022 at 11:51

Understood your point. What about that white space issue on the left side of the image, is there any possibilities for that? Or that too impossible? Please advise, Thanks. – Heyyou Jul 25, 2022 at 12:17

@Kishore: I have improved my answer a bit. – Axel Richter Jul 25, 2022 at 13:43

Thanks for explaining more in-depth. Please refer to the edit part in my question regarding header and footer. – Heyyou Jul 27, 2022 at 14:51

@Kishore: It is named `HeaderFooterType` because it is header **and** footer type. If there is a header/footer for the first page, then that first page will use first page header **and** first page footer. So that first page will **not** use the default page footer. – Axel Richter Jul 28, 2022 at 4:49