

# PdWriter Class

## ***Writing Paragraphs***

The PdWriter class (com.baseprogramming.pdwriter.PdWriter) is a class that demonstrates how to use the Apache project PDFBox. More so, it demonstrates how PDFBox can be extended to provide a more user-friendly interface to write content to PDF--without having to worry about breaking up a large chunk of text such that it fits in a page.

This class attempts to emulate a basic word processor approach, where text is written in paragraphs(PdParagraph class), and each paragraph has settings such as font and spacing.

The PdWriter class has two constructors

```
public PdWriter(PDDocument document, Margin margin)
```

And

```
public PdWriter(PageMetadata meta, PDDocument document)
```

The Margin class stores margin information (Top, Left, Bottom, and Right). The margins are stored as a PdUnit. The concept of a PdUnit (as with all other code in this project) is an experimental concept; its goal is to provide a client with a wide range of options for units of measures. Currently, the available units of measure are: PdInch, PdMillimeters, PdPica, PdPixels, and Points. All units of measure convert the given value to points--the standard unit of measure in graphic systems; the PdPoints class merely echos the value given.

The PageMetadata class has basic page information(Margin and PDRectangle), and has methods to compute page boundaries. The default PDRectangle is PDRectangle.LETTER

To get started with the PdWriter class, create create an instance:

```
Margin margin= new Margin(0.75f, 0.2f, 0.5f, 0.25f);
PdWriter writer= new PdWriter(pdDoc, margin);
```

Then create one (or more) PdParagraph objects:

```
PdParagraph heading=writer.createParagraph();
heading.setFont(PDType1Font.TIMES_BOLD);
heading.setFontSize(24);
heading.setAboveSpacing(new PdInch(0.75f));
heading.setBelowSpacing(new PdInch(0.75f));

PdParagraph body = writer.createParagraph();
body.setFirstLineIndent(new PdInch(0.3f));
body.setBelowSpacing(new PdInch(0.17f));

PdParagraph code=writer.createParagraph();
code.setFont(PDType1Font.COURIER);
```

```
code.setBeforeTextIndent(new PdInch(0.5f));
code.setAboveSpacing(new PdInch(0.3f));
code.setBelowSpacing(new PdInch(0.3f));
```

Write text (paragraphs, by calling the method PdWriter.write(PdParagraph, String):

```
writer.write(body, "Write text (paragraphs, by calling the method
PdWriter.write(PdParagraph, String) : " );
```

Note that there is a write(String) method, that creates its own PdParagraph instance with the default values.

## ***Numbered and Bullet Lists***

It is also possible to write a numbered or bullet list--use the class PdList, which extends PdParagraph. Use the createNumberedPdList method of the PdWriter class to create list paragraph style:

```
PdList list= writer.createNumberedPdList();
```

Or

```
PdList list= writer.createBulletPdList();
```

to create a bullet list. The line of code:

```
writer.write(list, "Java", "C++", "Python");
```

generates the following numbered list:

1. Java
2. C++
3. Python

A List of String can also be passed

To create a bullet list, use the createBulletPdList method:

```
list=writer.createBulletPdList();
```

and call the PdWriter write(PdList,...) method as with the numbered list:

\*Java  
\*C++  
\*Python

## ***Printing Tables***

It is also possible to print a data table. This is a more complex process, and requires the uses of three additional classes:

1. PdColumn
2. PdTableHeader

### 3. PdTable

The PdColumn class holds basic column information: label, name (for later reference and matching to the row data), and width. The PdTableHeader class contains the column definitions (PdColumn list) as well as header font information. Finally, the PdTable, which extends PdParagraph, contains the table header as well as additional border information. This class is used by the PdWriter class to write the table--given the data set. At the moment, the data asset must be passed as a list of maps, with the map representing the a row, with keys that correspond to the name in the PdColumn entries in the PdTableHeader. A PdTable instance can be created by calling one of the helper methods in the PdWriter class:

```
PdTable table = writer.createTable("First Name", "Middle Name", "Last Name", "D.O.B", "Memo");
```

Or

```
PdTable table = writer.createTable()
```

The columns must be created and added later when using this method.

The table exterior borders, cell padding, cell spacing, row border, column borders can be specified as follows:

```
Borders border= new Borders(3, 1, 2, 1);
table.setCellPadding(new PdPoints(10));
table.setRowBorder(1);
table.setColumnBorder(1);
table.setBorder(border);
```

Consider that there exists a method called getDataTable(int) that generates the list of maps needed to supply the table data:

```
List<Map<String, Object>> data=getDataTable(10);
```

The column widths can be calculated automatically by calling the PdTable method calculateColumnWidths:

```
table.calculateColumnWidths(data, 5);
```

This method uses row count specified (5 in this example) to calculate an adequate column width based on the column data size. After having calculated, individual column widths can be updated as follows:

```
PdTableHeader header=table.getHeader();
header.setFont(PDType1Font.TIMES_BOLD);
PdColumn memo=header.getColumn("Memo");
memo.setWidth(new PdInch(2));
```

Finally, the table can be printed as follows:

```
writer.write(table,data);
```

First Name	Middle Name	Last Name	D.O.B	Memo
Laree	Ginette	Barjenbruch	2005-05-11	Sieglar Marion city (pt.) commerce Gothenburg city

Ima	Devin	Liberatore	1968-10-18	genus bougainvillea Rost township bluebeard exploitive
Kelsey	Ngan	Quaife	1955-05-25	Menorrhagia revivification family psophiidae majuscular
Calista	Fallon	Saldi	1956-06-13	Cola MacIntosh nitromersol plagiarized Duleba
Rolanda	Diedre	Chartraw	2009-03-09	posadas multiplicative group Simkins Musich
Annika	Charles	Freidkin	1951-08-04	Swanson Zan Marietta city
Lezlie	Kathie	Ausland	1920-08-23	brawl Belayneh slezsko Innocent V warning coloration
Samella	Launa	Prum	1974-03-22	ministering unlocalising Webster County cover up Staples city (pt.) Niles city (pt.)
Lauryn	Darcel	Antista	1922-08-06	Fayette village Balance of Monroe township Lemoore city Woodworth town Pettiway Bruno city
Brook	Naida	Mandry	2006-05-15	Eagleville town (pt.) hyphenize Whitmore township

## Printing Images

Though not much work is saved, the PdWriter class also has two methods to write an image. This line of code:

```
writer.drawImage(new File("c:/tmp/moon1.png"), body, 240, 240);
```

prints the following image--resized to 240x240



while

```
writer.drawImage(new File("c:/tmp/moon1.png"), body);
```

prints the following image--using its actual size. the source image is actually quite large, and does not fit in the page--and thus only part of the image is printed



Generally, it would be best to resize the image to a more acceptable size

## Writing HTML

The PdWriter class also has rudimentary HTML writing capabilities. Consider the following snapshot (partial) of an HTML file:

```
<html>
  <head>
    <title>PdWriter HTML Input Sample</title>
  </head>
  <body>

    <p>This is a sample HTML file that can be printed using <b>
       This feature is still fairly incomplete. It can be node
       These values can be supplied in the <b>style</b> of an F
       see class <span style="font-weight: bold; font-family: c
    </p>

    <h1 style="color: #0000ff">This is an <i>H1</i> heading. </
    <h2 style="color: green">This is an <i>H2</i> heading. </h2>
    <h3 style="color: red">This is an <i>H3</i> heading. </h3>
    <h4>This is an <i>H4</i> heading. </h4>
    <h5>This is an <i>H5</i> heading. </h5>
    <h6>This is an <i>H6</i> heading. </h6>

    <p id="par1" class="first-par-style" style="padding-top: 10px">
       Notice that the tag name (<b>H1</b>) is italicized, whi
    </p>
    <hr style="border-top: 1px solid black; margin-top: 10px;" />

    <div style="border: 1px solid black; padding: 5px; margin-top: 10px">
      <p>
        This paragraph demonstrates a paragraph tag with a spar
        Such tags may pose problems--further complications at J
      </p>
    <hr style="border-top: 1px solid black; margin-top: 10px;" />

      <p>
        
      </p>
    </div>
  </body>
</html>
```

The output looks as follows:

---

## PdWriter HTML Input Sample

This is a sample HTML file that can be printed using **PDFBox** and the **PdWriter**. This feature is still fairly incomplete. It can be noted that the **PdParagraph** class uses the values above/below spacing and before/after indent to control spacing around the block of text. These values can be supplied in the **style** of an HTML element via **padding**. I have considered using the **Box Model** as in HTML, see class **com.baseprogramming.pdwriter.model.BoxModel**. However, I have not yet implemented this yet.

# This is an *H1* heading.

## This is an *H2* heading.

### This is an *H3* heading.

#### This is an *H4* heading.

##### This is an *H5* heading.

###### This is an *H6* heading.

Notice that the tag name (**H1**) is italicized, whilst it is in bold face on this explanation. This text content is on the **div** tag itself. How should be be treated. How about **span** tag content; or should a **div** be a paragraph, and a **span** just text--perhaps with formatting of its own within a paragraph. How about a **p** tag found within a **div**. Should it be implicit that the **div** sets the style for a group of paragraphs

It can be noted from the previous paragraph (which appears below text included directly on a div) that it is written on top of the previous text--there is nothing forcing a new line on text directly on a div. **This may indicate that text should not be expected (supported) directly on a div tag. The user may do so at own peril.**

This paragraph demonstrates a paragraph tag with a **span** tag within. The **span** tag could be used to provide additional formatting .

*This is an example of a bold tag inside an italics tag*



Note that the image above was printed correctly--using the *width* and *height* attributes specified in the **img** tag. It was assumed that the image (**Chrysanthemum.jpg**) was located in the same directory as the HTML file itself.

---

The file contents were written with the following line of code

```
writer.writeHtml(new File("c:/tmp/html-input-simple.html"));
```

It is also possible to pass an HTML snippet as follows:

```
writer.writeHtml("<p style=\"font-style: italic; padding-left: 10px;\">This  
is a paragraph printed with the <b>writeHtml</b>(String) method.</p>");
```

The output is:

*This is a paragraph printed with the **writeHtml** (String) method.*

Currently, Jsoup and cssparser are used to write HTML.

It is also possible to include an HTML table. However, minimal style attributes are considered at the moment. See sample below

Sample Table

First Name	Middle Name	Last Name	D.O.B	Memo
Shelley	Tijuana	Hanis	2000-07-27	Visby Jamestown town (pt.) mongo fils nardoo toffee-apple nondiscursive
Ernestina	Sherry	Chalcraft	1911-10-02	legally Berhley Edam pachyrhizus tuberosus kiwi Peguero osteophytic Milton city Wise County attingence
Vida	Inger	Beyser	1946-09-20	Perry township storklike unprating harmonica
Nicolas	Dewayne	Dalfonso	1921-05-16	Desroches Canal Fulton city photoelectric cell Kroese Kochville township lithotrite nonvibrating Pen Argyll borough predaylight inefficaciously
Dong	Clemmie	Bethune	1990-04-23	Lafavor in full swing Dippery dropped scone Burnside town unpoignantly
Gwenda	Jenee	Brumwell	2015-08-29	Choteau city full-face
Kristie	Tijuana	Pachter	1979-04-14	sterilisation columba Askiew Bergey Jamili unrelated

The following image shows a partial view of the HTML markup from which the table above was generated

```


| Shelley   | Tijuana | Hanis     | 2000-07-27 | Visby Jamestown town (pt.) mongo fils nardoo toffee-apple nondiscursive |
|-----------|---------|-----------|------------|-------------------------------------------------------------------------|
| Ernestina | Sherry  | Chalcraft | 1911-10-02 | legally Berhley Edam pachyrhizus tuberosus kiwi Peguero osteophytic Mil |
| Vida      | Inger   | Beyser    | 1946-09-20 | Perry township storklike unprating harmonica                            |


```

Admittedly, the output is not exactly purdy, but it does illustrate the potential.