



year 1  
2010  
project

# Android PDF Writer

# Description

**Android PDF Writer (APW)** is a simple Java library to generate simple PDF documents in Google's Android devices released under the BSD license.

The project page is: <http://coderesearchlabs.com/androidpdfwriter>

To learn more about how to use it, check the [Usage](#) section of this documentation.

There is a twin project for RIM's BlackBerry devices: the [BlackBerry PDF Writer \(BPW\)](#) with the same features than this one. Also, both versions has minimal dependencies and none of them outside Java and System libraries:

- for Android:
  - `java.util.ArrayList` (in Body and List)
- for BlackBerry:
  - `java.util.Vector` (in Body and List)
  - `net.rim.device.api.util.NumberUtilities` (in CrossReferenceTable)

## NOTE:

I wrote this library because I couldn't find a library to write PDFs in Android and I needed one to complete a weekend project.

So, the current APW library was coded in a weekend (May 14-16, 2010) following the PDF 1.4 Reference (I think my implementation is compatible with lower PDF versions also), so expect limited functionality: it handles only 1 Page and provides limited functionality to it (writes text and paint lines... well, it covers pretty well the needs that my weekend project had at least, and may be it also address the needs of many other android projects out there with simple reporting needs).

## UPDATE:

Six months after that (Nov 15-16, 2010) I'm adding two things: first an overload to `setPageFont()` with an encoding parameter due to a user requirement for writing special characters in text. Check the PDF 1.4 Reference APPENDIX D, to be sure which encoding fits better for your needs, since as you can see there are symbols that have no octal value in all of the encodings. And second, an overload to `addText()` with a text transformation parameter, thanks to Jon from Gigaram Technologies for the suggestion.

There are so many things that can be added (images, filters, etc) and enhanced (it does not handle key/values for instance, etc etc etc), so if you add something that is useful for you (I think this library is a good base to be extended, check the class structure), please let me know since it might be helpful for others too.

Finally, by April 2011, I ported it to BlackBerry.

Enjoy,  
Javier Santo Domingo  
[j-a-s-d@coderesearchlabs.com](mailto:j-a-s-d@coderesearchlabs.com)

# Usage

**APW** is very simple to use, it offer an very straightfoward class to give you full control of the simple PDF document you are creating.

Look at this example:

```
private String generateHelloWorldPDF()
{
    PDFWriter mPDFWriter = new PDFWriter();
    mPDFWriter.setPageHeight(400);
    mPDFWriter.setPageWidth(320);
    mPDFWriter.setPageFont(
        crl.android.pdfwriter.StandardFonts.SUBTYPE,
        crl.android.pdfwriter.StandardFonts.TIMES_ROMAN,
        crl.android.pdfwriter.StandardFonts.WIN_ANSI_ENCODING
    );
    mPDFWriter.addRowContent("1 0 0 rg\n");
    mPDFWriter.addText(70, 50, 12, "hello world");
    mPDFWriter.addRowContent("0 0 0 rg\n");
    mPDFWriter.addText(30, 90, 10, "© CRL", crl.android.pdfwriter.
        StandardFonts.DEGREES_270_ROTATION);
    mPDFWriter.addLine(150, 150, 150, 200);
    mPDFWriter.addRowContent("[] 0 d\n");
    mPDFWriter.addRowContent("1 w\n");
    mPDFWriter.addRowContent("0 0 1 RG\n");
    mPDFWriter.addRowContent("0 1 0 rg\n");
    mPDFWriter.addRectangle(40, 50, 20, 50);
    String s = mPDFWriter.asString();
    return s;
}
```

To learn more about the library, check the [The Library](#) section of this documentation.

# The Library

The main class of this library is **PDFWriter**, which presents the following public members.

Page related methods:

- **void** setPageHeight(**int** Value)
- **void** setPageWidth(**int** Value)
- **void** setPageFont(**String** subType, **String** baseFont)
- **void** setPageFont(**String** subType, **String** baseFont, **String** encoding)

Content related methods:

- **void** addRawContent(**String** rawContent)
- **void** addText(**int** leftPosition, **int** topPositionFromBottom, **int** fontSize, **String** text)
- **void** addText(**int** leftPosition, **int** topPositionFromBottom, **int** fontSize, **String** text, **String** transformation)
- **void** addLine(**int** fromLeft, **int** fromBottom, **int** toLeft, **int** toBottom)
- **void** addRectangle(**int** fromLeft, **int** fromBottom, **int** toLeft, **int** toBottom)
- **String** asString()

All the font and text options are included in the **StandardFonts** class, presenting the following variety of public members.

Subtype constants:

- **String** SUBTYPE

Font constants:

- **String** TIMES\_ROMAN
- **String** TIMES\_BOLD
- **String** TIMES\_ITALIC
- **String** TIMES\_BOLDITALIC
- **String** HELVETICA
- **String** HELVETICA\_BOLD
- **String** HELVETICA\_OBLIQUE
- **String** HELVETICA\_BOLDOblique
- **String** COURIER
- **String** COURIER\_BOLD
- **String** COURIER\_OBLIQUE
- **String** COURIER\_BOLDOblique
- **String** SYMBOL
- **String** ZAPDINGBATS

Encoding constants:

- **String** MAC\_ROMAN\_ENCODING
- **String** WIN\_ANSI\_ENCODING

Rotation constants:

- **String** DEGREES\_0\_ROTATION
- **String** DEGREES\_45\_ROTATION
- **String** DEGREES\_90\_ROTATION
- **String** DEGREES\_135\_ROTATION
- **String** DEGREES\_180\_ROTATION
- **String** DEGREES\_225\_ROTATION
- **String** DEGREES\_270\_ROTATION
- **String** DEGREES\_315\_ROTATION

# License

Android PDF Writer  
Copyright (c) 2010-2011, Javier Santo Domingo (j-a-s-d@coderesearchlabs.com).

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Tools

In the development of **Android PDF Writer** the following tools were specifically involved:

for source code editing

**Eclipse Foundation's Eclipse**

<http://www.eclipse.org>

for source code compiling

**Google's Android SDK**

<http://developer.android.com/sdk>

When working at **Code Research Laboratories** the following tools are used:

in the test management field

**Gurock Software's TestRail**

<http://www.gurock.com/testrail/>

in the version control field

**VisualSVN Ltd's VisualSVN**

<http://www.visualsvn.com/>

in the similarity analysing field

**RedHill Consulting's Simian**

<http://www.redhillconsulting.com.au/products/simian/>

in the application lifecycle management field

**Inedo's BuildMaster**

<http://www.inedo.com/buildmaster/>

in the documentation field

**EC Software's Help & Manual**

[http://www.ec-software.com/products\\_hm\\_overview.html](http://www.ec-software.com/products_hm_overview.html)

# History

DATE	DESCRIPTION
2010.05.14	started coding
2010.05.15	improvements
2010.05.16	more improvements
2010.05.19	initial release
2010.11.15	added setPageFont() overload with encoding parameter
2010.11.16	added addText() overload with text transformation
2011.04.20	ported to BlackBerry (creating the BlackBerry PDF Writer twin project)
2011.04.25	added addRectangle()
2011.05.19	added Documentation