

# Adding PDF support to your application

---

**Leonard Rosenthol**  
**Senior Software Engineer**  
**Digital Applications, Inc.**



# Overview

---

- **Libraries**
- **Tools**
  - PDF Creation
  - PDF Rendering & Printing
  - PDF Data Extraction
  - Other
- **Inside a PDF**



# How I do things

---

- **A copy of the final document is on Digital Application's website (<<http://www.digapp.com>>) for your downloading pleasure**
- **Although I've left time at the end for Q & A, I'm more than happy to take questions at any time.**



# Libraries

---

- **PDFlib**
- **ClibPDF**
- **retepPDF**
- **iText**
- **Etymon™ PJ**
- **Adobe PDF Library**
- **SPDF**



# PDFlib

Thomas Merz

---

- **Pros**

- available for all major platforms and languages
  - Mac OS, Windows, Unix, AS/400
  - C, C++, VB/COM, Java, Perl, Python & TCL
- thread safe
- supports all major features of PDF 1.3
- Unicode-savvy (bookmarks, annotations & content)
- source is available (except ActiveX version)

- **Cons**

- creation only
- only supports the Base14 fonts + Base 7 CJK (w/o external files)



# ClibPDF

FastIO Systems

---

- **Pros**

- available for all major platforms
  - Mac OS, Windows, Unix
- built-in support for 48 Roman + 7 CJKV fonts
- high level “graphing” primitives
- thread safe
- supports all major features of PDF 1.3
- source is available

- **Cons**

- creation only
- only available as a C API



# retepPDF

retep/Peter Mount

---

- **Pros**

- Written in Java
- Directly integrates with Java printing API's
- open source
- free

- **Cons**

- Written in Java
- creation only
- limited font support
- no support for images



# iText

Bruno Lowagie

---

- **Pros**

- Written in Java
- open source
- free

- **Cons**

- Written in Java
- creation only
- only supports text (no images, etc.)





# Etymon™ PJ

Etymon

---

- **Pros**

- Written in Java
- Can read & modify existing PDFs
- full source available

- **Cons**

- Written in Java
- hasn't been updated in a while, support requires \$\$
- limited font support
- no support for images



# Adobe PDF Library

Adobe Systems

---

- **Pros**

- It's the “standard”!
- Includes a few extras
  - PDF rendering
  - PDF -> PS/EPS
  - Text Extraction

- **Cons**

- Not currently available!
- Available only for Mac OS and Windows
- FDF support requires a separate library
- Not thread safe
- Leaks memory



# SPDF

## Digital Applications

---

- **Pros**

- Available for Mac OS, Windows, Unix and IBM System 390
- Uses the same API's as Adobe PDFL & Acrobat SDK
- Source available
- Includes a few extras
  - PDF rendering
  - PDF -> PS/EPS
  - Text Extraction
  - FDF Parser
- Thread safe, no memory leaks
- Debugging version tracks client object allocation/usage

- **Cons**

- No support for PDEdit calls



# PDF Creation

---

- **Direct from your Application**
- **PDFMarks**
- **Printer Drivers**
- **Converting from other format**



# Direct from your Application

---

- **Hello World (PDFL/SPDF)**
- **Hello World (PDFlib)**



# Hello World (PDFL/SPDF)

```
PDDoc thePDDoc = ( PDDoc )NULL ;

// create a new document
thePDDoc = PDDocCreate( ) ;

// add some document info
char *authStr = "Leonard Rosenthol", *creaStr = "LibTester";
PDDocSetInfo( thePDDoc, "Author", authStr, strlen(authStr) );
PDDocSetInfo( thePDDoc, "Creator", creaStr, strlen(creaStr) );

// add a page
PDPage thePDPage;

// create the page
FixedRect mediaBox;
mediaBox.left = fixedZero; /* dimensions of page */
mediaBox.top = Int16ToFixed(4*72); /* in this case 5" x 4" */
mediaBox.right = Int16ToFixed(5*72);
mediaBox.bottom = fixedZero;
thePDPage = PDDocCreatePage( thePDDoc, PDDocGetNumPages(thePDDoc) - 1, mediaBox );

// add some content
PDFUtils::AddTextContent( thePDDoc, i );

// bookmark it!
char pageStr[50];
sprintf( pageStr, "Page %d", i+1 );
PDBookmark thePDBookmark = PDFUtils::NewBookmarkToPage( thePDDoc, i, pageStr );

// all done, let's release it!
PDPageRelease(thePDPage);

// Save the output PDDoc
PDDocSave( thePDDoc, PDSaveFull | PDSaveCollectGarbage, theOutputASPathName, NULL, NULL, NULL );

// Close the PDDoc
PDDocClose( thePDDoc );
```



# Hello World (PDFlib)

---

```
PDF_boot();

// create a new document
p = PDF_new();
if (PDF_open_file(p, filename) == -1) {
    return 1;
}

// set document info
PDF_set_info(p, "Title", "PDFlib test program");
PDF_set_info(p, "Creator", "PDFtest");

// start the page & bookmark it
PDF_begin_page(p, a4_width, a4_height);
(void) PDF_add_bookmark(p, "Centered text", 0, 1);

// add some centered text
textfont = PDF_findfont(p, "Times-Roman", ENCODING, 1);
PDF_setfont(p, textfont, (float) (1.5 * FONTSIZE));
PDF_show_boxed(p, title, CENTER, y, 0, 0, "center", "");

// end the page
PDF_end_page(p);

// close the document
PDF_close(p);
PDF_delete(p);
PDF_shutdown();
```



# PDFMarks

---

- **A little Postscript is good for the soul**
- **What can you do?**
- **Putting them into your output**
- **How hard is it?**
- **Making a PDFMark**
- **For more information**





# A little Postscript is good for the soul

---

- **The pdfmark operator is a PostScript extension which is only implemented in Acrobat Distiller (and similar tools).**
- **Using this operator, many non-layout related features of a PDF file can be defined in the original document or in the corresponding PostScript code.**



# What can you do?

---

- **Notes**
- **Links**
- **Bookmarks**
- **Article threads**
- **Named destinations**
- **General document information**
- **Viewer properties**
- **Viewer preferences**
- **Page transitions**
- **Creating form fields**
- **Inserting JavaScript instructions**
- **and lots more!!**



# Putting them into your output

---

- **PICT Comments**

- an “extension” mechanism in the PICT format which allows PICT creators to provide extra info to renders (including the LW driver)

- **You put the PS code into a standard PS comment using well documented and well tested technology**

- this is how many applications provide PS alternative code to QD instructions
    - rotated text
    - hairline vectors



# How hard is it?

---

```
pcPostScriptBegin      = 190;
pcPostScriptEnd        = 191;
pcPostScriptHandle     = 192;
pcPostScriptBeginNoSave = 196;

void PxSendPostScriptText(char *inText, short inLen)
{
    Handle          h = NewHandle( inLen );

    if ( h ) {
        PicComment(pcPostScriptBegin,0,NIL);
        BlockMoveData(theText, *h, n);
        PicComment(pcPostScriptHandle, n, h);
        PicComment(pcPostScriptEnd,0,NIL);

        DisposeHandle(h);
    }
}
```



# Making a PDFMark

---

```
/pdfmark where {pop} {userdict /pdfmark /cleartomark load put} ifelse
[  /Title (Untitled)
   /Author (Eric Raymon)
   /Subject (The Cathedral and the Bazaar)
   /Keywords (Linux is cool)
/DOCINFO pdfmark
```

% The following code defines a bookmark:

```
[  /Page 1
   /View [/XYZ 44 730 1.0]
   /Title (Start)
/OUT pdfmark
```



# For more information

- 
- **PDFMark Primer by Thomas Merz**  
<[http://www.pdflib.com/pdfmark/pdfmark\\_primer.pdf](http://www.pdflib.com/pdfmark/pdfmark_primer.pdf)>
  - **PDFMark Reference Manual, Adobe Systems**  
<<http://partners.adobe.com/asn/developer/acrosdk/DOCS/pdfmark.pdf>>
  - **PDF Reference Manual, Adobe Systems**  
<<http://partners.adobe.com/asn/developer/acrosdk/DOCS/pdfspec.pdf>>



# Printer Drivers

---

- **PrintToPDF**
- **Acrobat Distiller**
- **Ghostscript**



# PrintToPDF

James W. Walker

---

- **Supports any application that can print!**
- **Supports CJKV**
- **Supports auto-conversion of URL's, bookmarks, and more**





# Acrobat Distiller

Adobe Systems

---

- **It's the “standard”**



# Ghostscript

Aladdin Enterprises

---

- **Supports PDF 1.3**
- **Supports standard PDFMarks**
- **Open Source**



# Converting from other format

---

- **HTMLDoc**
- **SVG2PDF**
- **FOP**
- **XEP**



# HTMLDoc

Easy Software

---

- **Open Source**
- **Only supports HTML 3.2 (+ some extensions)**
- **Lots of special formatting options**



# SVG2PDF

Digital Applications, Inc.

---

- **Technology Preview**



# FOP

apache.org

---

- **XML/XSL Formatting Objects (FO's) -> PDF**
- **Written in Java**
- **Open Source**
  - must compile/build yourself
- **Under development, limited functionality**



# XEP

RenderX.com

---

- **XML/XSL Formatting Objects (FO's) -> PDF**
- **Written in Java**
- **Source is available**
  - must compile/build yourself
- **Unsupported - “evaluation only”**



# PDF Data Extraction

---

- **Xpdf Tools**
- **Ghostscript**
- **PDFDecoder**





# Xpdf Tools

FooLabs

---

- **Open Source**
- **AppleScriptable app for**
  - document information
  - text extraction
  - image extraction
  - PDF->PS
  - PDF->HTML



# Ghostscript

Aladdin Enterprises

---

- **Open Source**
- **AppleScriptable app for**
  - text extraction
  - PDF->PS



# PDFDecoder

LimitPoint Software

---

- **Scriptable application for**
  - document information
  - text extraction



# PDF Rendering & Printing

---

- **jPDF**
- **XpdfTools**
- **Ghostscript**



# jPDF

ReallItaly

---

- **Written in Java (requires Java2)**
- **Designed for use on web pages**
- **Development seems to have halted?**



# XpdfTools

FooLabs

---

- **AppleScriptable tool for rendering pages to raster images**



# Ghostscript

Aladdin Enterprises

---

- **Open Source**
- **Incorporates a PDF renderer (via PDF->PS), but not exported via AS**



# Other

---

- **Combining & Splitting documents**
- **Adding new content (“Stamping”)**
- **Forms processing**
- **Batch security apply/remove**
- **The skies the limit!**





# Inside a PDF

---



# Conclusion/Wrap Up

---

- **Go out there and do cool stuff!**



Q & A

---

Q & A

