Introduction to PDF Programming

Leonard Rosenthol
Lazerware

Overview

- What might you want to do with PDF?
- Review of available libraries
- Review of the PDF file format
- Developing with the Acrobat API
- Developing with PDFlib

You are here because…

- You're a programmer looking to expand in doing stuff with PDF.
- You're already programming PDF using some library and wanted to hear about other libraries.
- There wasn't anything else interesting to do.
- You're a friend of mine and wanted to heckle

How I do things

- You should all have copies of the presentation that you received when you walked in.
- There is also an electronic copy of this presentation (PDF format, of course!) on my website at http://www.lazerware.com/
- I've left time at the end for Q&A, but please feel free to ask questions at any time!

What to do with PDF?

- Creation
  - Report generation
  - Content repurposing
  - Document Conversion
- Manipulation
  - Adding text or images
  - Form filling
  - Append or removing pages
  - Imposition
  - Adding structural elements
    - Bookmarks, hyperlinks, etc.
  - Securing and signing

What else can you do?

- Imaging
  - Printing
  - Rasterization (conversion to bitmap)
- Content extraction/conversion
  - Text, HTML, XML
  - Postscript

Review of Libraries

- Creation Only
  - PDFlib
  - ClibPDF (FastIO)
  - Panda (StillHQ)
  - PDF File Creator (FyTek)
  - PDF in a Box (Synactis)
  - PDFever (Perl Script Studio)
  - SanFace PDFLibrary (SanFace)
  - ReportLab

Libraries (cont)

- Creation Only
  - retepPDF (Peter Mount)
  - Root River Delta (Root River Systems)
  - The Big Faceless PDF Library (Big Faceless)
  - iText (Lowagie)
- Creation & Manipulation
  - PDFLibrary (Glance)
  - Life*JOVE (Corena)
  - PJ (Etymon)
  - activePDF Toolkit (ActivePDF)

Libraries (cont)

- Imaging
  - 5D PDFLibrary (Global Graphics)
  - Ghostscript (Artifex)
- Everything
  - Acrobat SDK
  - Adobe PDFLibrary
  - DocuCom PDF Core Library (Zeon)
  - SPDF (Appligent)

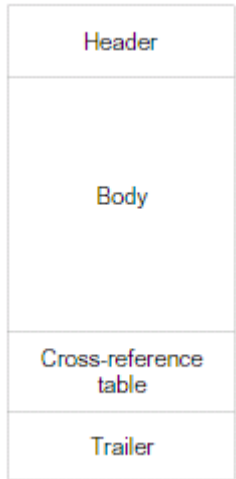
What’s in a PDF?

Peeling the layers of PDF

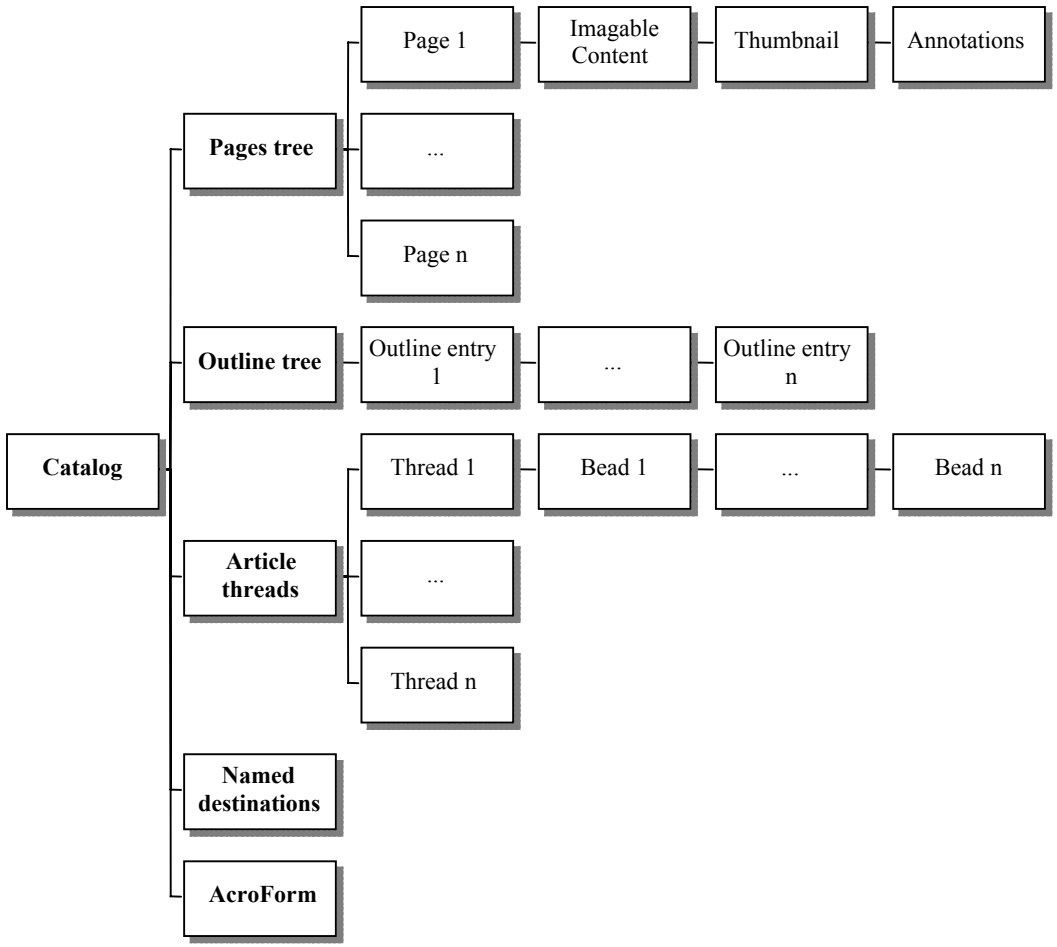
- PDF file
  - physical container in a file system containing the PDF document and other data
- PDF document (aka page description)
  - Contains one or more pages, where each page consists of text, graphics and/or images as well as hyperlinks, sounds, etc.
- “other data”
  - PDF version, object catalog, etc.

PDF Document Layout

- Header
  - Specifies PDF version
- Body
  - Sequence of objects
- XREF
  - Where to find each object
- Trailer
  - Tells where to find XREF



Structure of a PDF document



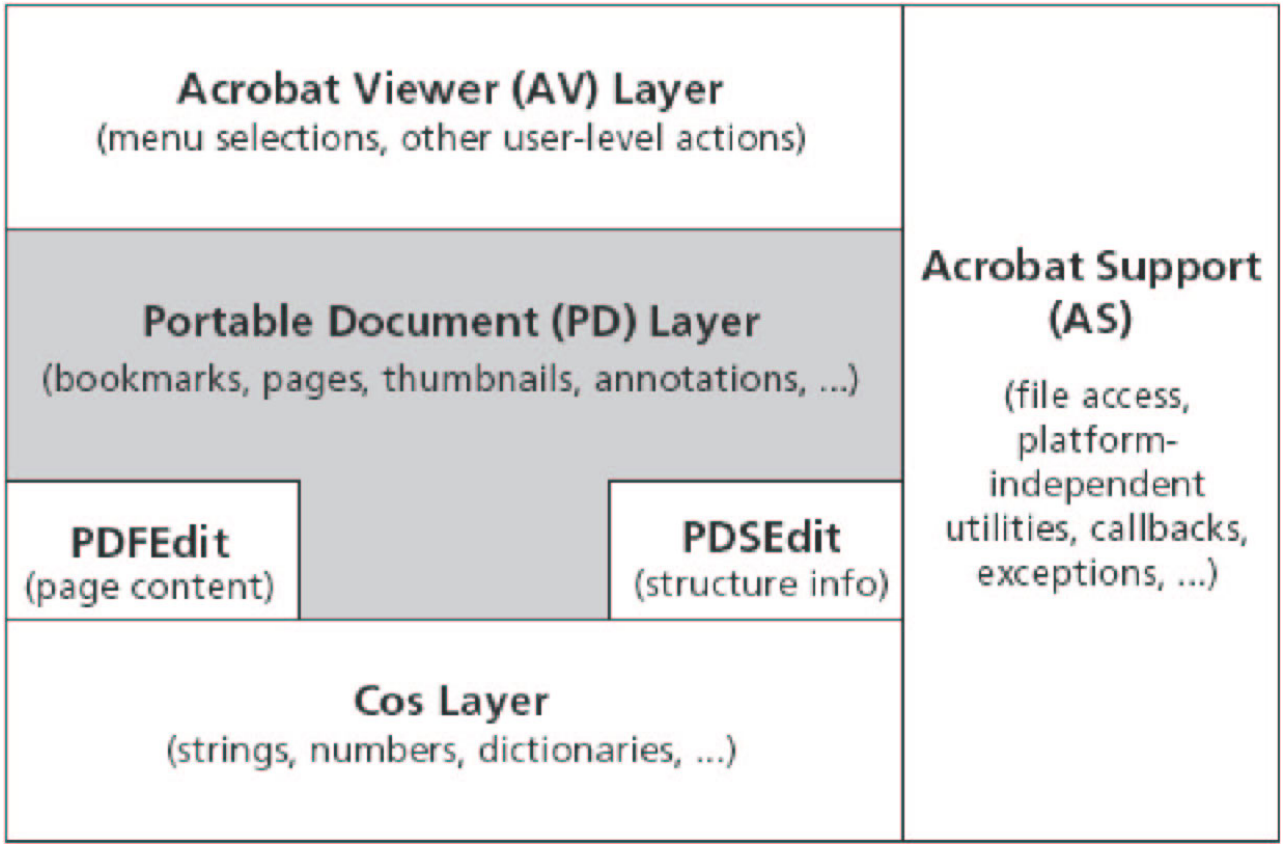
Smallest PDF

```
%PDF-1.1
1 0 obj
<<
/Pages 3 0 R
/Type /Catalog
>>
endobj
2 0 obj
<<
/Type /Page
/Parent 3 0 R
>>
endobj
3 0 obj
<<
/Kids [ 2 0 R ]
/Count 1
/Type /Pages
/MediaBox [ 0 0 612 792 ]
>>
endobj
```

```
xref
0 5
0000000000 65535 f
0000000015 00000 n
0000000085 00000 n
0000000136 00000 n
0000000227 00000 n
trailer
<<
/Size 5
/Root 1 0 R
/ID[<5181383ede94727bcb32ac27ded71c68><5181383ede94727bcb32ac27ded71c68>]

>>
startxref
277
%%EOF
```

A look at the SDK



Where to find the “SDK”?

- Acrobat Plugins
  - Mac OS & Windows
- Adobe PDFLibrary
  - Mac OS, Windows, Linux x86, Solaris
- SPDF (Appligent)
  - Mac OS, Windows, Linux (x86 & PPC), Solaris, AIX, HP/UX, Digital Unix, IBM System 390
- DocuCom PDF Core (Zeon)??
  - Windows

What's in there?

- Not every implementation of the "SDK" has 100% of the same features (even between Acrobat and PDFLibrary).

- Access to everything in a PDF file
  - Read, Add, Modify
- Content extraction
- PDF rendering
  - to bitmap or platform window
- Printing

Everything is an “object”

- CosObj
  - CosString, CosInteger, CosArray, CosDict
- PDDoc
  - PDPage, PDBookmark, PDAnnot
- AVDoc
  - AVWindow, AVPageView, AVTool
- PDEObject
  - PDEText, PDEImage, PDEPath

PDF Objects

- Acrobat treats the objects as opaque, while SPDF lets you view their contents in the debugger (incl. objectID!)
- All objects are NOT created equal!
  - PDDoc != AVDoc != CosObj
- Although Acrobat allows you to use them interchangeably, SPDF does not and in fact will generate compile time errors
  - PDDoc == CPDDoc, CosObj == CCosObj
  - But there are API calls to go between them
    - PDDocGetCosObj()

ASAtoms

- Rather than working with literal strings all the time, many SDK calls take ASAtoms.
- Think of them as a list of name/values pairs which are keyed by strings.
  - improved memory management & ease of use
  - As such, many developers use a single set of global ASAtom variables.
    - SPDF even includes macros for doing this
- ASAtomFromString()
- ASAtomGetString()
- ASAtomExistsForString()

Fun with File Systems

- ASFileSys
  - A base “class” the represents a way for the SDK to read & write the data of a PDF file. (a fancy Stream)
  - Acrobat provides only file-based ones
  - SPDF also provides memory, FTP & HTTP
- ASPathName
  - ASFileSysCreatePathName (const ASFileSys fileSys, ASAtom pathSpecType, const void* pathSpec, const void* mustBeZero);
  - ASPathFromPlatformPath(void* platformPath)

Error Handling

- DURING/HANDLER/ENDHANDLER
  - In Acrobat itself, these map to something akin to setjmp/longjmp
    - Trying to mix them with C++ exceptions can be a problem.
    - You can't nest them!
  - SPDF actually defines them as try/catch blocks
  - ERRORCODE

More on Error Handling

- Unfortunately, Acrobat does NOT always “throw”. Sometimes you have to use other methods
  - foo == NULL, PDxxxIsValid(), etc.
- CosNull != NULL
  - If want a null CosObject, you can call CosNewNull() to get one. BUT that should be treated as a valid object and NOT as NULL.

Error Handling Sample

```
DURING
    theASPathName = ASPathFromPlatformPath( inPDF ) ; // Create the ASPathName
    thePDDoc = PDDocOpen( theASPathName, NULL, NULL, true ) ; // Open the PDDoc
    if ( thePDDoc == (PDDoc)NULL ) {
        fprintf( gOutputFile, "# Unable to open PDF file - %s\n", inPDF ) ;
        ASRaise ( ASFileError( fileErrOpenFailed ) ) ;
    }
HANDLER
    theError = ERRORCODE ;
    if ( theASPathName != NULL ) {
        ASFileSysReleasePath( NULL, theASPathName ) ;
        theASPathName = ( ASPathName )NULL ;
    }
    ASGetErrorString( theError, theAcrobatMessage, sizeof( theAcrobatMessage ) ) ;
    fprintf( stderr, "# Error: %s\n", theAcrobatMessage ) ;
    return ;
END_HANDLER
```

Thread Safety?

- Acrobat, nor the Adobe PDFLibrary, are thread safe! As such, you should not try to use them in a threaded environment OR make your own threads outside the SDK.
  - There are some exceptions to this rule if you are VERY careful, but you're playing with fire.
- SPDF comes in both thread safe and non-thread safe versions.
  - If you know you don't need threads, then why take the performance overhead!

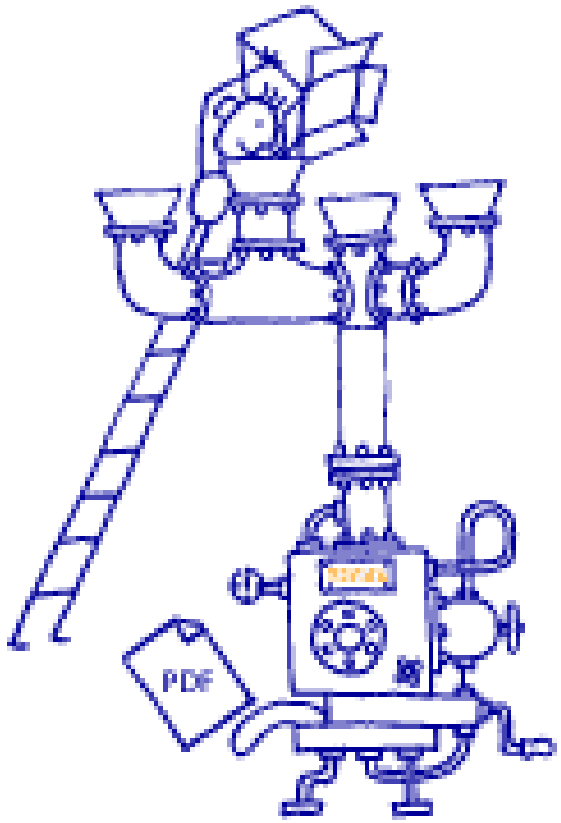
SPDF Memory Tracker

SPDF object usage table:

	created	freed	leaked	high water mark
Array	17	17	0	16
HashTable	4	4	0	4
HashtableEntriesTable	5	5	0	4
ASAtom	145	145	0	124
ASFile	1	1	0	1
CosArray	4	4	0	4
CosBoolean	0	0	0	0
CosDict	5	5	0	5
CosDoc	0	0	0	0
CosDocRevision	1	1	0	1
CosName	23	23	0	23
CosNull	1	1	0	1
CosNumber	6	6	0	6
LZWFilter	0	0	0	0
FlateFilter	0	0	0	0
PDBookmark	1	1	0	1
PDBead	0	0	0	0
PDDoc	1	1	0	1
PDPage	2	2	0	1
PDPath	0	0	0	0
PDFileSpec	0	0	0	0
PDFont	0	0	0	0

Splitter Example (SDK)

PDFlib



What's in there?

- PDF Creation/Generation
  - Text, images, vectors, bookmarks, links, etc.
- Allows importing of pages from other PDF's as "XObjects" with accompanying PDI library

- Accessible from C/C++, Java, Perl, PHP, etc.
- Available as an ActiveX/COM component
- Available as platform-neutral C source

Everything is a PDF?

- You initialize PDFlib and get back a reference to an opaque “PDF” structure.
  - PDF *p = PDF_new();
- Each “PDF” can have only a single PDF open at any one time for generation, BUT you can have as many “PDF”’s around as you want (eg. One per thread).

Error Handling

- Each language binding uses it's native error handling mechanism
  - Eg. C++ & Java == exceptions
- For C, you can specify a function to be called
  - Provides you with the type/class of error and a string describing it.
  - You decide whether a given error is fatal or can be ignore (more of a "warning")
- You can also specify globally how you want to deal with warnings (treat as errors or not).

Hello (PDFlib)

Q & A