

Introduction to developing with PDF/Acrobat & 3rd Party Libraries

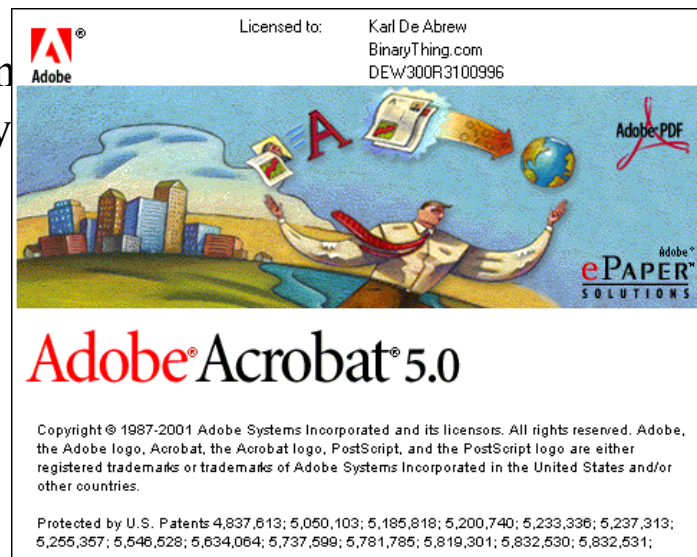
Abstract:

- Introduction to PDF development using Acrobat 5.0 SDK and third party libraries

Karl De Abrew, CEO of
BinaryThing.com
incorporating Planet PDF

After Party:

<http://forum.planetpdf.com/>



Overview of presentation

- Why you would want to develop for PDF?
- What are your options?
 - Acrobat SDK
 - Plug-ins
 - IAC applications
 - Forms Toolkit
 - Third Party Libraries
 - Roll-your-own
 - Combination of the above
- Frequently Asked Questions
- Where do I find out more?

30 minute session, 10 minutes of questions



Why would you need to develop for PDF/Acrobat?

- PDF is the defacto standard for distribution of final form documents, a recent push in the few years has seen it also vying for contention as
 - the ebook format of choice and a
 - Strong contender in the forms market.
- You obviously wouldn't be here if you didn't see significant potential in PDF.
- Whilst PDF and Acrobat represents an out-of-the-box solution to many problems, it has the potential to solve many more.
- What is lacking in these situations is you -- the software developer.
- What I aim to give you during this presentation is an independent perspective on the types of problems and solutions that you might deploy in integrating PDF in your environment.



Why would you need to develop for PDF/Acrobat? Cont...

- The PDF specification is freely available and any developer (given time) can develop a product that acts as a PDF consumer or creator.
- A pre-existing PDF creator and PDF consumer, Adobe Acrobat, is extensible from both the inside out, and the outside in via its plug-in and IAC APIs.
- Don't reinvent the wheel --- there are over 500+ Acrobat-related tools that we have counted at our Planet PDF Web site – <http://www.planetpdf.com/tools>
- Know the base of platforms --- In a recent interview with Adobe CEO Bruce Chizen, he quoted that Acrobat represents 20% of Adobe's revenue, and that 90% of the product sold is Windows-based.



What are your options?

1. Acrobat SDK
2. Third Party Libraries (non-Adobe)
3. Roll-your-own
4. Some combination of the above...



Acrobat 5 SDK – what is it?

- The Adobe Acrobat note states that
 - “...The Acrobat SDK is a “toolbox” that contains header files, type libraries, simple utilities, sample code, and documentation. These tools help you design and develop Acrobat plug-ins as products or as part of integration with your application. The plug-ins control the Acrobat product suite, using Interapplication Communication (IAC) interfaces such as Apple events, OLE, and DDE.”



What Do I Get the SDK from?

You can download the installer from the Web site to obtain the entire Acrobat SDK, or you can download each technical note separately. The sample files and documentation are what comprise the Acrobat SDK.

<http://partners.adobe.com/>

Note: The Forms Toolkit is now included in the Acrobat 5 SDK



Getting Started with the SDK

- The Acrobat 5 SDK is basically divided into three areas. Documentation and files relating to:
 - Plug-ins
 - IAC (plug-outs) OLE/COM/DDE
 - Forms Toolkit (recently incorporated)
- There are a plethora of documents contained within the SDK. It can be daunting to come to terms with these all in one sitting.
- You will have to determine what it is you want to do and then determine if there is a sample that gets you pointed in the right direction.
- Take a divide and conquer approach to your development. By having a wholistic understanding of what you can do with Acrobat first, and then by being aware of what's possible and what's not, then you'll be able to concentrate your learning efforts in the most appropriate area.



Demonstration 1

- Examine SDK folders



What Is a Plug-in?

- An Acrobat plug-in is a program written in ANSI C/C++ that uses the Acrobat API and adds functionality to Acrobat or Acrobat Reader. A plug-in program file goes into a plug-in folder/directory (usually named “Plug-ins”) and is initialized upon startup of the Acrobat viewers.
- Plug-ins are dynamically-linked extensions to the Acrobat viewers. They can hook into the user interface in a number of ways and can register to be called when a variety of events occur in the application.
- Plug-ins are:
 - DLLs under Windows
 - Code fragments on Macintosh PowerPC
 - Shared libraries under UNIX



Demonstration 2

- Page Extractor - Plug-in
 - Adds menu-item and popup dialog box
 - Extracts all pages from the active document



What is IAC?

- IAC stands for InterApplication Communication. It's name comes from the Apple world and it is implemented as AppleScript on the Mac and OLE/COM/DDE on the Windows-client.
- The number of objects and methods available are limited when compared with the plug-in world, however, as IAC allows you to connect other applications to Acrobat it can be highly useful.
- In fact, if you require an IAC method that Acrobat does not provide, then you can write your own plug-in to act as an OLE server that can provide this.



Demonstration 3

- Page Extractor - IAC application
 - Extracts all pages from the active document



Unsupported activities by ASN

- Use of any Acrobat product in a multithreaded way.
- Use of any Acrobat product as a server process accessed by multiple clients,
- Use of Distiller as an NT service.
- Use of the “Portable Document Format Reference Manual” to develop a third-party application that writes PDF files without the use of Adobe Acrobat products.
- Interapplication Communication (IAC) between a plug-in and a third-party application.
- Use of platform API methods or API methods of applications other than Acrobat.
- Use of the ActiveX Control or Netscape plug-in to display a PDF file in an external application besides Internet Explorer or Netscape.
- Automating importing image files using the Import and Capture plug-ins to Acrobat.



Third-Party Libraries

- Third party libraries have been built by a wide-range of companies to cater for a wide-range of problems.
- Third party libraries can be general purpose and Acrobat-API compatible such as Appligent's SPDF Library. Or specific in their purpose and built for a specific environment such as a direct to PDF package that renders XSL formatting objects.
 - [ClibPDF](#) - FastIO Systems
 - [PDFlib](#) - Thomas Merz Consulting and Publishing
 - [Data2PDF](#) - SANFACE Software
 - [Simple-PDF](#) - Tangent Software
 - [PDF Library](#) - Adobe Systems
 - [retepPDF](#) Java Generator - Peter T Mount
 - [FOP](#) - An Open-Source XSL Formatter and Renderer
 - [PDF Library and Toolkit](#) - GLANCE
 - [JClass](#) PageLayout
 - [SPDF Library](#) - Appligent (formerly Digital Applications)
 - [PDF Core Library](#) - Zeon Corp.



Roll your own

- The PDF Specification is published and freely available from <http://partners.adobe.com/>
- If you do not find an existing product that suits your needs, or a library that you can integrate or extend, then you may like to explore this route.
- Note: PDF is not trivial – many tasks require more than a substantial amount of work and thus it is often not cost-effective to follow this path. However, in some cases, for whatever reason, circumstances will dictate that this is the most appropriate choice. For whatever reason, you must be the judge of this.



Demonstration 4

- Create your PDF postcards with PDFlib



Some combination of the above

- Obviously, and in fact most often, when you work in the real world, things don't fit as neatly into boxes as we'd like them.
- Hybrid solutions can be an effective way of balancing the time-old buy vs. build argument and allow you to avoid reinventing the wheel and retain control over your project.



Example situations

- Importing a library of TIFF files on-the-fly, setting document metadata, securing them, and finally emailing them to an address as specified in a database.
- Separating a PDF into its constituent pages.
- Converting a collection of Word files to PDF
- Extracting the content from PDF on-the-fly, converting it to text
- Building your own search engine plug-in.



Example situations cont...

- Viewing a PDF file in your own application's window using Acrobat.
- Using Acrobat to implement a context-sensitive help system.
- Writing a plug-in to add or change menu bars, menu items, or tool-buttons (that is, customizing the user interface).
- Controlling an Acrobat or Reader session externally
- Adding annotation types.
- Adding action types (such as going to a URL in a Web browser).



Frequently Asked Questions

- How do I develop a plug-in for Reader?
- Can I create a plug-in for Reader to save?
- What happened to the Visual Basic plug-in SDK?
- What happened to the Adobe PDF Library?
- How do I get official support from Adobe?



More information

- For more information on developing with Adobe Acrobat head to the following resources:
- Planet PDF
 - <http://www.planetpdf.com/developer> (samples)
 - <http://www.planetpdf.com/tools> (wheels)
 - <http://forum.planetpdf.com/> (developer conference)
- Adobe Systems
 - <http://partners.adobe.com/> (download the SDK)

