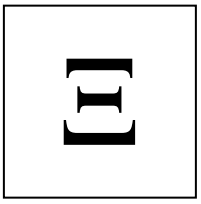
PDF made easy

A short introduction



By Forest White

Date: *November 2001*

Version: *1.00*

Forest White, Technology Consultant

Document History

Version	Date	Changed By	Change Description
.80	November 12, 2001	Forest White	Initial Draft

Table of Contents

1 PDF made easy

This document is intended to give a logistical and technical overview of Portable Document Format. There is more information here than can be used by the average user. If you only wish to get started quickly, skip to the section that details your operating system and

1.1. Introduction

A lot of documents these days can be found in a PDF file. The documentation for the Java 1.1 class libraries, an abstract about mixing and isosurface geometry of turbulent transverse jets, the IRS tax forms, and even Professor Emily Bender's Instruction Manual for her class on Syntactic Theory are available on the web in Portable Document Format. But few sources exist to give you an idea of what PDF is or how to use it. After searching Internet resources, I realized that only one source had specific information on this subject – Adobe Systems Inc. They provide a manual describing the concepts of the format, their application software, Adobe Acrobat, and one-year memberships to view their API (not including the Acrobat source code) for no less than $395. Adobe Systems keeps a tight grip upon this technology because of its versatility and platform independent strategy.

This document is designed to provide a quick overview of the PDF technology and how to use it for average and technically skilled users. This technology is gaining popularity in the governmental and academic spheres of our society so it is useful to understand why and how it does what it does.

1.1.1. Acrobat Products

Adobe Acrobat is the only application that is widely available for viewing, creating, and modifying a PDF file. It does not come as one neatly versatile application, however. Different versions of this software are produced by Adobe to suit different needs. In actuality, each product is a different "shell" of the same Acrobat Kernel, with different linked libraries and binary files distinguishing one from another. The .exe file for each product contains the directions for which set of functions to use. Using an .exe that does not correspond to your installation results in a crash unless the files associated with that product are installed correctly.

This functionality causes certain problems. One common problem occurs when the standard edition and reader are installed on the same machine. If one is in use and the user attempts to start the other, usually the OS will simply make the currently running application the active one and not open the other. This can lead some users to believe that their application is not working correctly because when the reader is open, other versions do not seem to run.

It is sometimes important to determine which application is running. In most cases, the appearance of the application can allow a user to distinguish one from another. The reader application is shown in Figure 1. The small feature set and the name Acrobat Reader in the title bar distinguish it.

Each product comes with its own set of functionality but without any of them, a PDF file cannot be used. They are as follows:

- The most accessible viewer application is the Acrobat Reader. It allows the user to view, search, navigate, and print PDF files. The standard version comes with a plug-in for most Internet browsers so that Acrobat reader can be used within a browser context. It is readily available on the World Wide Web from Adobe as well as a number of associated business partners.

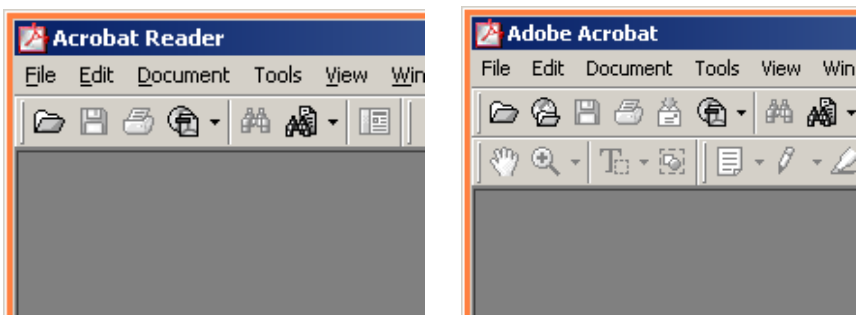


Figure 1. The Acrobat Reader and Acrobat Standard Editions

- A small upgrade to the Acrobat Reader, the standard edition of Acrobat (simply named) provides the ability to create navigational links, add annotations to the PDF file, set security policy, and edit it in some limited ways. The name in the title bar and the added functionality will distinguish this version from its reader counterpart, as shown in figure 1. It is important to note that this version cannot create PDF files. In fact, PDF files are never created by and Acrobat application. PDF is a format, not a desktop publishing application.

- Acrobat Distiller creates PDF files from postscript files. The relationship between postscript and PDF will be described in more detail in the *PDF and PS* section. Its user interface, if opened directly, is very different from the reader and standard edition of the application. It is shown in figure 2. When distiller is installed, it will also install a startup client, called Acrobat Assistant, that will allow you to print to PDF from selected applications.

- Acrobat Writer is a largely unseen application that provides plug-ins to a large variety of desktop publishing applications and business tools. It does not use the intermediary step of postscript like distiller and is useful to the user who is interested solely in creating documents in a user-friendly fashion.

- Acrobat Capture is a plug in for the Writer software that allows the user to create text searchable PDF documents from scanned images.

There are few other options open to anyone seeking to utilize PDF files. Some applications are available from the open-source community and several other corporations have produced new applications with API. To find out more

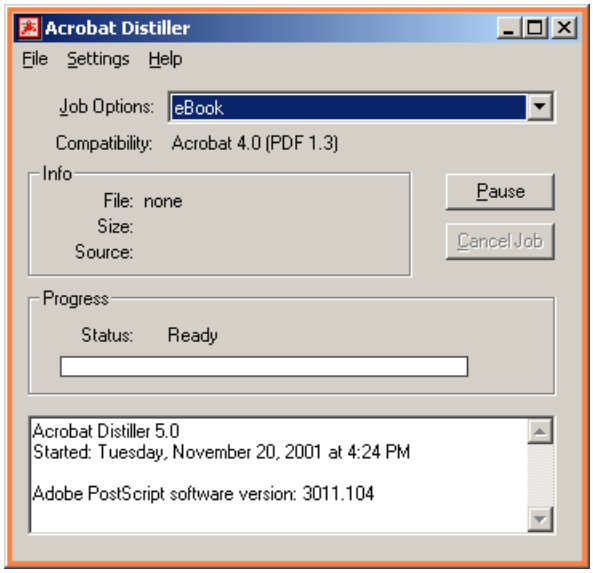


Figure 2. The Acrobat Distiller 4.0 interface

about the tools available on your operating system, consult the section that details it.

1.1.2. Interpretive Solutions

A current trend in software solution technique involves what has come to be called Interpreted Markup. This basically describes data formats and language styles that are interpreted as they are used rather than being compiled, assembled, or otherwise made into machine code before the user can activate them. Interpretive Markup is the guiding concept used in HTML, XML/XSL, Java, Perl, AWK, postscript, and PDF. When the user accesses a file of interpreted Markup, an associated application, called a context application, starts and "interprets" the markup, executing it without the necessity of a compiler. An interpretive solution has the following advantages:

1. Interpreted Markup can be transported as source code. Although this is not actually true of Java and PDF, these two markups retain the small file size associated with source transport. Although the ability to "hand edit" Java and PDF is limited, AWK, Perl, HTML, postscript, and XML/XSL files are completely modifiable in their standard forms.

2. Interpreted Markup is suitable for the inexperienced user as well as the programming guru. With appropriate instructions, any user level may use Interpreted Markup files as long as the appropriate context application has been integrated into their operating system. The must stunning example of this would be the standard web (or, more accurately, HTML) browser. It is useful to the web programmer as well as the elementary school student.

3. Interpreted Markup is platform independent. As long as context applications are made for each operating system, interpreted markup files will be interpreted in exactly the same way (theoretically) on any platform. This means that developers can use any computer to create the files and any consumer can pick up software products and information without having to wonder if there is a version for the operating system they prefer.

PDF provides most of the advantages of an Interpretive Solution. Its context application, Adobe Acrobat, is supported for the 3 major operating system families that are in use today – Microsoft Windows, Macintosh OS, and most UNIX platforms. With an Acrobat Reader, anyone with a working computer can view and print a PDF file.

There are some specific features of PDF that should be noted. PDF does not use an open source markup. It is actually a binary file, which is similar in some ways to a compiled solution. However, with the Acrobat standard edition, writer, or distiller, PDF files can be edited on any platform. PDF also has its own security policies. This allows for version control and rights management

that is independent of the Operating System. This security feature makes it especially appealing to corporations and governmental organizations.

1.1.3. PDF and PS

PostScript is a programming language that is designed for printing graphics and text without consulting any device features. Any printer designed to print from the postscript standard will produce the same result as any other. This level independence was a very limited interpreted solution where the context application was embedded in the printer driver and supported by the printer hardware. It has been upgraded from the original version created by Adobe in 1985 to a more robust tool that is used in by most printer manufacturers today. A free software context application is available, called ghostveiw or ghostscript, that can be used to view the postscript file in its interpreted format without printing it.

PDF is similar to postscript and uses postscript to hold a portion of its information. PDF was created with the same imaging model as Postscript. This is particularly useful because this means that PDF can easily be converted to postscript (and thus, easily printed) and back again with most of the Acrobat applications.

A good analogy that may explain the similarities between the two is the way that PDF and postscript create images and text. The page descriptions in both formats draw a page by placing "paint" on a defined area (the page). The paint may form many shapes including fonts (letter shapes), vector drawn curves and areas, and digitally defined or sampled images. The paint may be of any color (or so Adobe claims) and the paint is strictly 2-dimensional, meaning that when one applies "paint" over another painted object, the paint overwrites any paint that was previously occupying that space. Objects are not defined independently, but rather in the page context.

The primary difference between the postscript implementation and PDF implementation is the manner of editing each type. PostScript is generally edited by editing the markup, either directly or through an application that can rewrite it with a postscript printer driver. PDF has no readily accessible markup and is therefore edited with the context application. This application manipulates the binary data and the markup associated with its organizational data.

A PDF file has several other advantages over postscript that make it a more versatile solution for modern applications. It supports objects such as hypertext links that are useful for interactive viewing. It has a strictly defined file structure that supports streamlined and random searching of specific pages within a file. It also is able to store its own font libraries to ensure font retention.

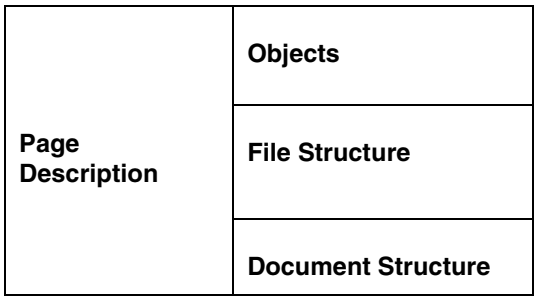


Figure 3. The Components of a PDF file

A PDF file is composed of four parts as shown in Figure 3. The first part, the Objects component, contains all of the objects (text, graphics) that are to be displayed in the document. All of the objects in this component have predefined types that are analogous to the object types in postscript. PDF gives fixed names to its object variables for every document whereas postscript allows the programmer to name their own variables. This makes the context application fairly universal and decreases the time needed to interpret the file. Any new objects created for a PDF file will be given a name and included in the Objects component.

The second part is the PDF file structure. This component determines how the contents of the PDF file are stored and accessed. You may think of it as a table of contents with memory mapping. It creates the overall file structure, which allows the context application (Acrobat) to access the Objects within the PDF file.

The PDF Document Structure organizes the Object content to be displayed. It can be likened to a style sheet for HTML or a XSL template for XML. It organizes Objects into pages, text threads, image frames, and the like and puts them into an order. It is also the component of PDF that tracks link destinations, forms created specifically by Acrobat to organize Objects into viewable formats, and fonts.

The last component is the PDF page description. This portion of the PDF document handles the "math" of the document. An object definition is not limited to simple ASCII text or binary data. A curve, for example, can be defined as a function just as in the field of mathematics. The object in this case would be comprised of a data type (curve x) and the coordinates for the beginning and end of it. The curve will then be calculated by a function in the page description and inserted where the Document Structure dictates. Many object functions, including text, shading, image, path, and link operators, reside in this component. Like the Objects in the Object Component, these functions are defined once and are only capable of calculating this display of an object. The operators are fixed but remain analogous to the more general operators of the postscript type.

1.2. Creating and Editing PDF Files

Creating a PDF file is actually creating a document first and then converting to PDF later. Most of you will have your favored Desktop Publishers already picked out. But whichever one you use, most of the principles described here are still applicable.

1.2.1. Windows

Windows is the most popular Operating System currently for home and office users. Adobe, despite their close working relationship with Macintosh, realizes the market need for products that work with windows. All of the Acrobat applications have window's versions, but some of the considerations of using windows to produce PDF files may surprise you.

1.2.1.1. Creating a PS file

The best way to convert your desktop publishing files to PDF and insure that the features of the document remain intact is to convert it to postscript first. This is an extra step in your PDF creation process that may save you unwanted problems later.

To change a file into PS, you will need to install a PS printer driver. Open your Printers folder from the Start menu under Settings (see figure 4)

Figure 4. How to find the Printers folder

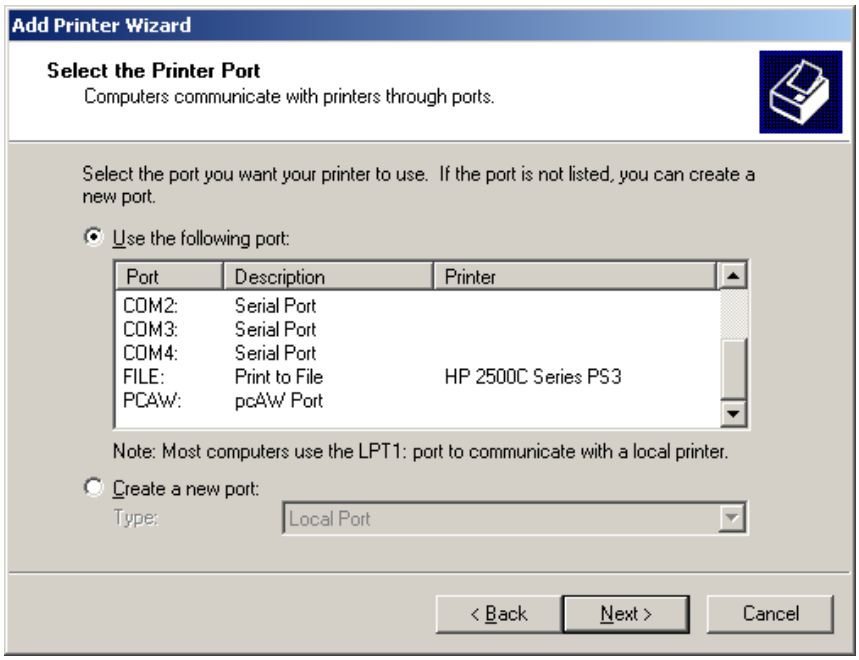


Figure 5. The FILE: option allows you to print to a file instead of a printer device

Click on Add Printer and select a local printer. You will be given the option of selecting the port the printer is to be installed on. Simply select FILE: so that the output of the printer driver will be a file (see figure 5). A list will be presented to you of different printer drivers that windows has available. I suggest choosing and HP printer but you must select a printer (of any brand) that has PS (PS2, PS3) in its name. If this is not present, the driver may not use PostScript and therefore cannot make PS files.

Once the printer is installed, you may select it when printing a document to create a PS file. It will ask you what you would like to name the file and where it should be similar to the way it would if you were saving it. Once the file is in PS format, you may use distiller to change it into a PDF file.

1.2.1.2. Distiller Methods

When it comes to changing your document into PDF, there are actually 2 options available to you. The distiller changes PS files to PDF and the writer will directly change any document of an approved format into PDF by using the print command. Recent distiller versions have also integrated this user-friendly feature, but it is important to understand what each product does so that you do not end up with a PDF file you cannot use.

The distiller is pretty straightforward. You load a PS file into it and it outputs a PDF file. Simple.

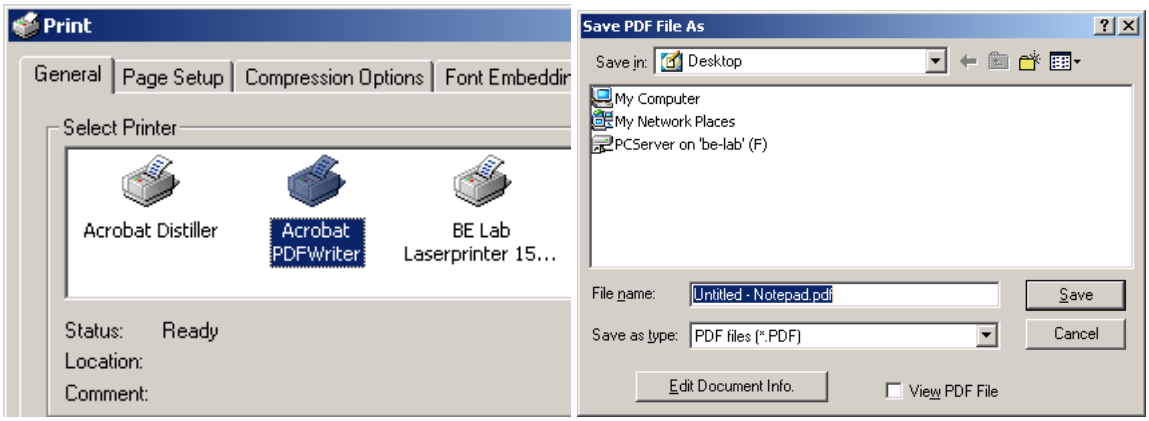
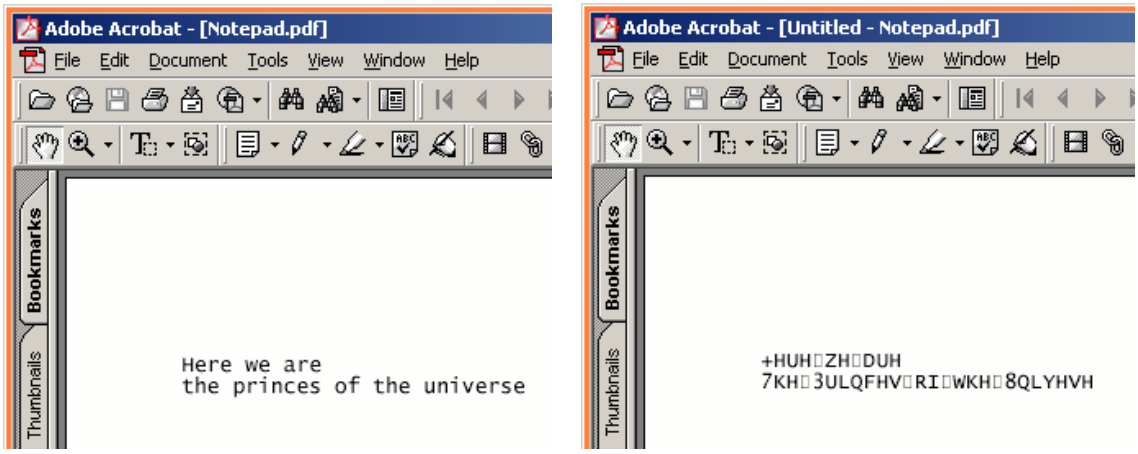


Figure 6. The PDF Writer and Distiller print options. They may look the same but they are not.

With the writer, you are not converting to PS before making PDF. Adobe created this option to make the process simpler for the end user. Figure 6 gives a screenshot of the printing options for a newer version of distiller and the current writer version.

PDF uses a Unicode (32-bit) encoding to manage its content. Because of this, older text editors, such as notepad, need to be saved in such a way that Unicode is written instead of ASCII. An ASCII text file that is not in Unicode or converted to PostScript (also a Unicode format) will not produce a useful document with Writer. Writer does not support non-Unicode documents. As you can see in figure 7, the simple ASCII file is saved as garbage when Writer creates it, but is preserved when distiller converts it into PS first.



1 2

Figure 7. The results of creating a PDF file from an ASCII file with Distiller (1) and Writer (2).

1.2.2. Mac OS

Macintosh has recently come to the attention of Adobe Systems Inc. by virtue of a business contract that was mutually beneficial to both companies. All of the Acrobat applications that are current are compatible with Macintosh OS 7 or better.

1.2.2.1. Creating a PDF or PS File

If your Macintosh is running OS X, then you are in luck. Any Carbon™ application that would normally run on OS X can save its contents as a PDF file (no printer driver needed). It is really that simple.

For users that have not or cannot upgrade to OS X, there remains the distiller and writer applications. PostScript has been a standard for Macintosh print drivers since the Apple Laser Writer (as in the first one). So you will need to install a PS printer driver on a Macintosh in almost the same way you would on a Windows machine.

First, you will need to install the PostScript® printer description (PPD) files that describe the fonts, paper sizes, resolution capabilities, and other features that are standard for your PostScript printer. PPD files are used by PostScript printer drivers to determine how to print your document. If you already have a printer installed that supports PostScript Printing (see the printer's documentation), then you can skip to the actual printing using your current drivers.

The PPD files are at www.adobe.com/support/downloads/main.html in the Printer Drivers section. You may also find the AdobePS general driver (no printer needed) there. Given that we will print to a file, this printer driver and its associated PPD will work well for our purposes. To install a PPD file, copy it to the System Folder > Extensions > Printer Descriptions folder. Install the AdobePS driver by double-clicking on the Installer icon. In the Installer, click Continue and then click Accept to accept the licensing agreement. The easy install installs the AdobePS, PrintingLib, Adobe Printing Library, AdobePS Read Me file, AdobePS License Agreement file, and a Printer Descriptions folder, containing three AdobePS extensions (DXTNs): Custom Page, Virtual Printer, and Watermark. Note that this folder contains no PPD files - they must be installed separately.

Once you have a PS printer driver properly installed with the appropriate PPD, you can create a PS file by selecting the print option from your desktop publishing application. A printer dialogue box will appear. And you must make certain selections for it to make a PS file properly. First, the printer must be your PS enabled printer driver. The destination must be "FILE" and the box

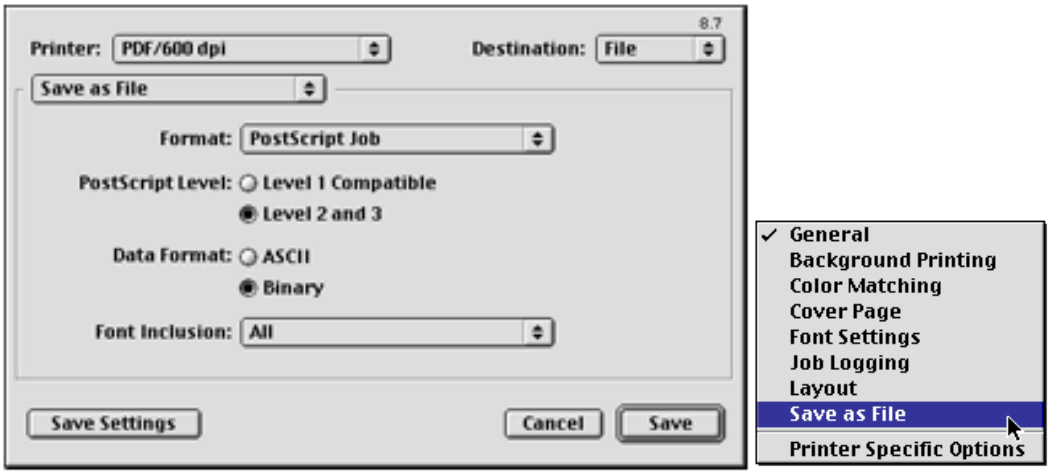


Figure 8. The Macintosh print dialogue box and the option that will provide you with PS options

under the Printer and Destination boxes must read "Save as File". At this point the printer dialogue box will change to resemble the one in figure 8. The Format box should be set to PostScript Job (see figure 9). One of the options for this box is also Acrobat PDF (assuming you have distiller installed). If you choose this format, the printer dialogue box will change again to provide you with distiller options. These options are more restrictive than the PS options and they only appear if you have distiller installed on the specific computer you are using. If none of these issues are a problem, you may simply create the PDF file and be finished.

Otherwise, there are a few more options to worry about with PostScript (see Figure 9). The Post Script level should be set to "Level 2 and 3" to give you faster loading and processing. The data format, assuming you are going to eventually change this file in to PDF, should be "Binary". Remember that PDF files are Unicode binary and distiller may not produce the desired result if it is creating a PDF document from an ASCII file. Use the "All" font inclusions option and then click the Save button. You will be prompted for a file name and location for your PostScript file. This completes the method for creating PS files.

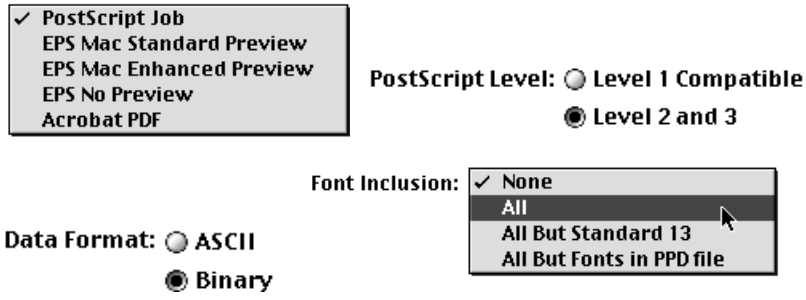


Figure 9. The various options to create a PS file that can easily be converted into PDF

1.2.2.2. Distiller Methods

Distiller in Macintosh works almost exactly like the Windows version. Simply put a PS file into the application and out pops a PDF file.

1.2.3. Editing

Editing a PDF file may be the most important parts of your document production process. Although Acrobat is not a word processor or graphics editor, it and place objects into and take objects out of a PDF file for other Acrobat applications to access.

1.2.3.1. Writer Basics

Among the basic things you will need to make use of the editing tools of Acrobat, the "grab" tool (which looks like a right hand) is probably the most useless. It is worth noting because it allows you to move objects around on the page. Placement can sometimes be an issue and it can move pages and other large objects with ease. This usually means that it will screw up your project by changing too much.

A more elegant selection tool is the text select (see figure 10) and graphics select tools. Anything selected by these tools can then be manipulated with the hand. It will take some time to get used to using these tools as they were not designed to do any serious editing, but rather to move PDF specific objects into place so that they do not obscure things that are already written.

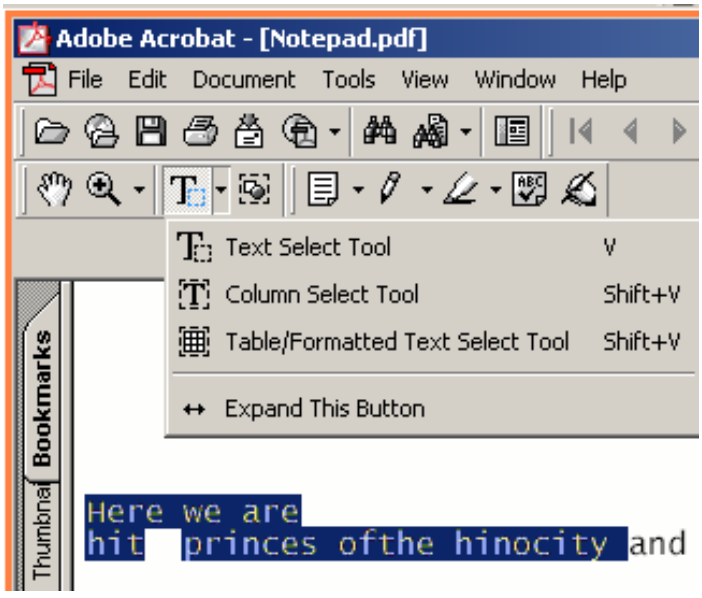


Figure 10. There are many choices when using the Text Select tool.

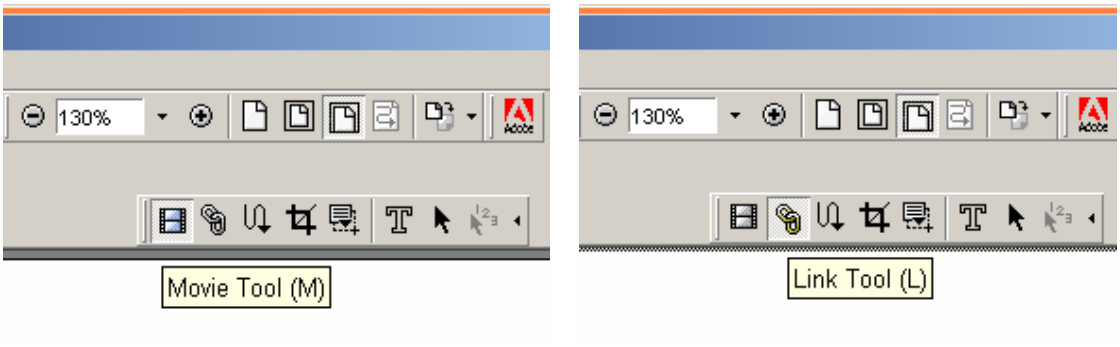


Figure 11. Interactive content can really spice up your document.

The zoom tool (magnifying glass with a + or – in it) allows you to zoom in and zoom out on a document page. Acrobat can only zoom out to 8% of the page size or zoom in to 3200% of an image size when viewing it.

1.2.3.2. Formatting Tools

Since PDF supports a lot of different object content, formatting can be essential to the type of document you are producing. Some of these tools are designed for use in presentations or web pages but they are still worth understanding.

In figure 11 we see two of the powerful interactive tools. The Movie tool supports mpeg and avi video compressed files and can place them in a square area of relatively any size. The link tool simply allows you to make an object or a portion of an abject into a hypertext or soft link.

The article tool, seen in figure 12, is also an interactive tool. It provides a way for readers to auto zoom on portions of a document. It creates boxes, which

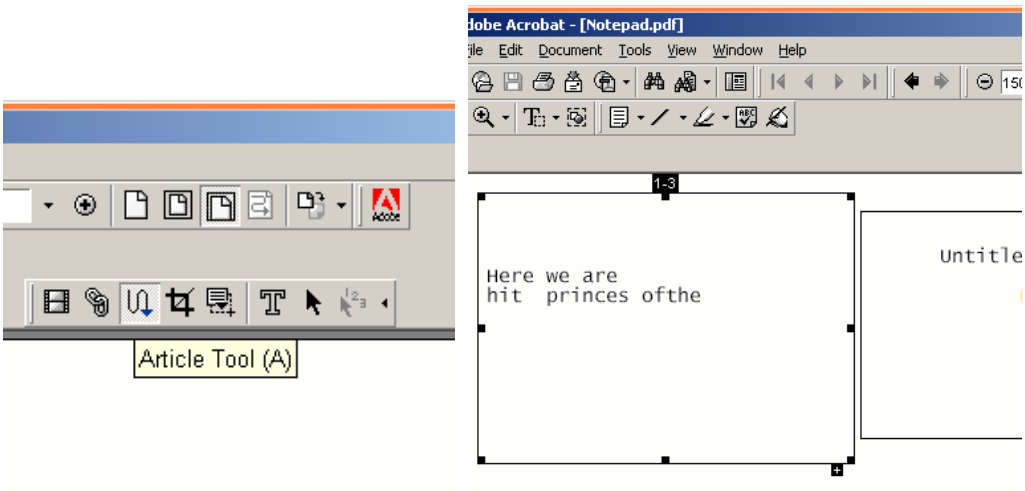


Figure 12. The article tool can alert your readers of things that are important

are not visible to the reader, that fills the area of the Acrobat window as if it had been zoomed in on precisely when selected during reading.

The remaining tools are less oriented on interaction and more on presentation. Activating the crop tool will give you the crop pointer. This pointer can manipulate graphics object by resizing them and cutting pieces out of them. It is probably most useful for formatting pages. If you double click on the background of the page with it, the crop pages window will be displayed. This gives you the ability to precisely change the size of your pages and how they will be displayed (see figure 13)

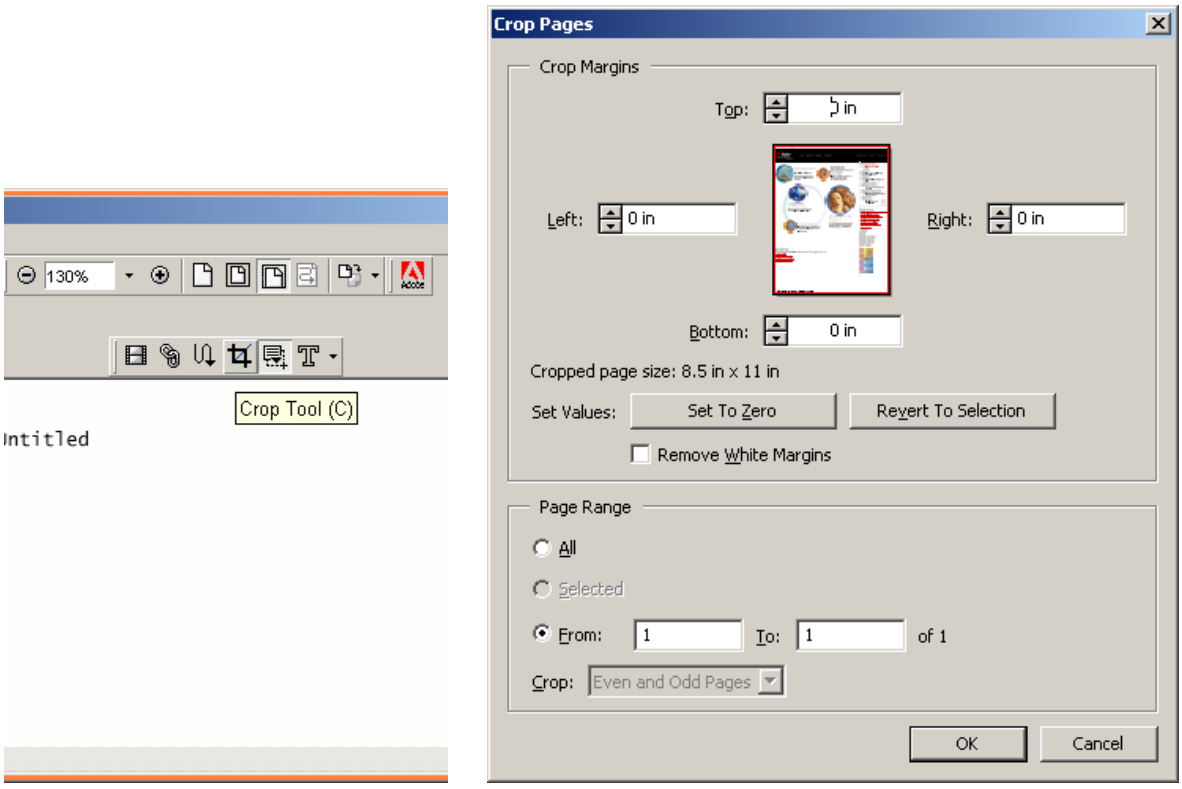


Figure 13. The crop tool and the crop pages window.

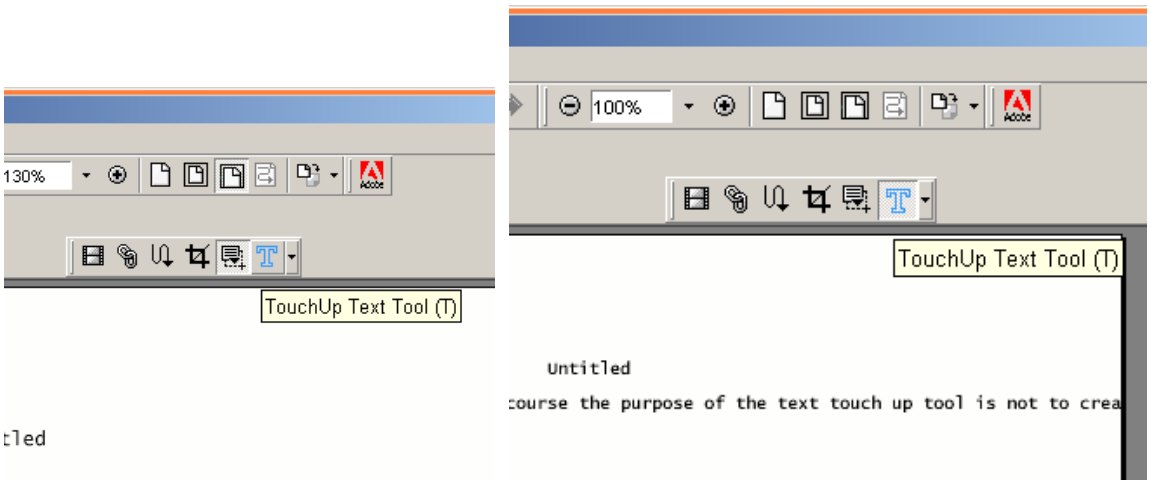


Figure 14. The text touch up tool allows you to edit text. Carriage returns are not valid characters and your text may run off of the page, however.

The text touch up tool is the only tool that actually allows you to manipulate the text of the page directly. A text object is composed of a field that surrounds the text. The text touch up tool allows you to access this field and add or delete contents from it. In figure 14, there is an example of the effects of such an edit that runs off of the page. Just as the page is an object on the canvas, so is the text. Text need not be fully on a page to be included in a file. The text touch up tool does not allow you to use carriage returns so you can create text that runs off of the page and therefore not displayed.

1.2.3.3. Content Tools

Content tools provide a means to put PDF specific objects into a PDF file. The objects they create are usually only useful to those viewing a page but a few will appear when the page is printed. When editing other people's documents, this is the most appropriate set of tools to use.

Largely, these tools contain pull down menus just to the right of the icon on the tool bar. They contain different things or a particular type. The defaults are the most useful for editing (in general) but the pull down manus can contain several useful items as well. It is worth your time to try all of the tools in each pull down menu to build your own style of editing.

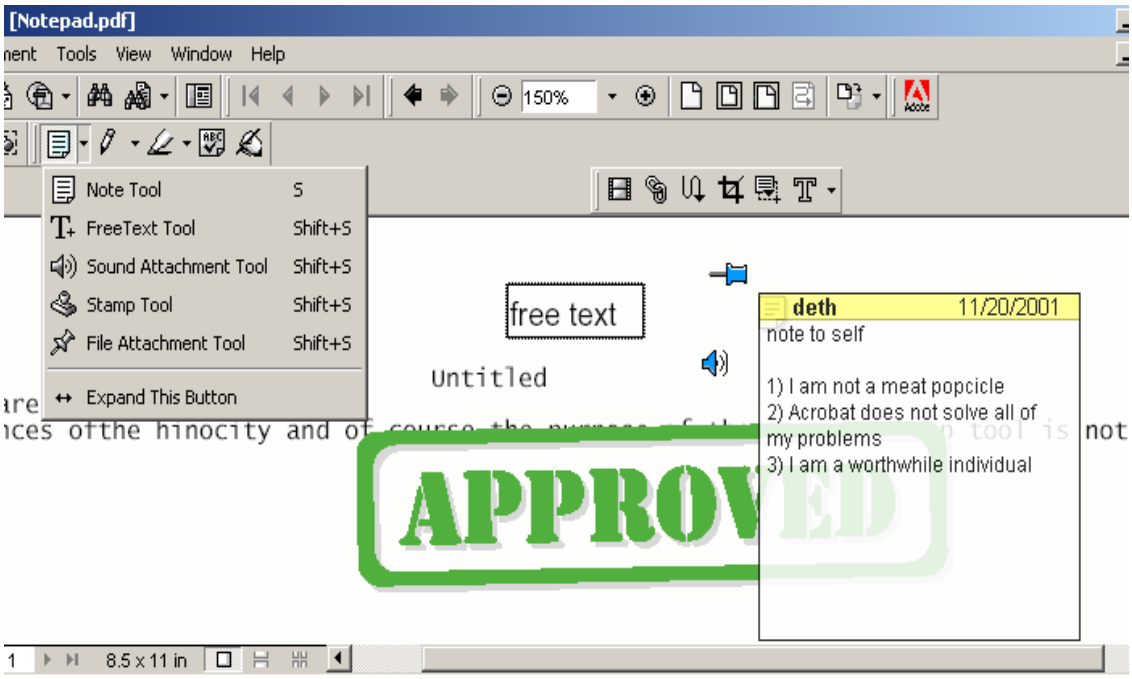


Figure 15. The "S" tools and an example of each.

The note tool and its derivatives (shown in figure 15) allow you to include encapsulated messages in several formats. The default, the note tool, places a small icon on the document that when selected opens up to a memo board. The author's name and the date appear at the top of the note and the text is entered by that same author. The free text tool allows you t create another text object that is actually on a page by itself with the only text viewable through the window you create when you use the tool. The sound tool allows you to record a message. It can be accessed by clicking on the speaker icon that is created when it is used. The stamp tool paints an image (it does not create a true graphics object) that places a stamp on the canvas. The default is the green "approved" stamp that is shown in figure 15. A file can be attached to the file and directly accessed via the blue thumbtack icon that it places on the canvas. This link is not handled by acrobat but instead calls on the operating system to handle it. If the file format is not recognized by the Operating System and the person viewing the PDF file does not have the prop, then it will not be able to be opened.

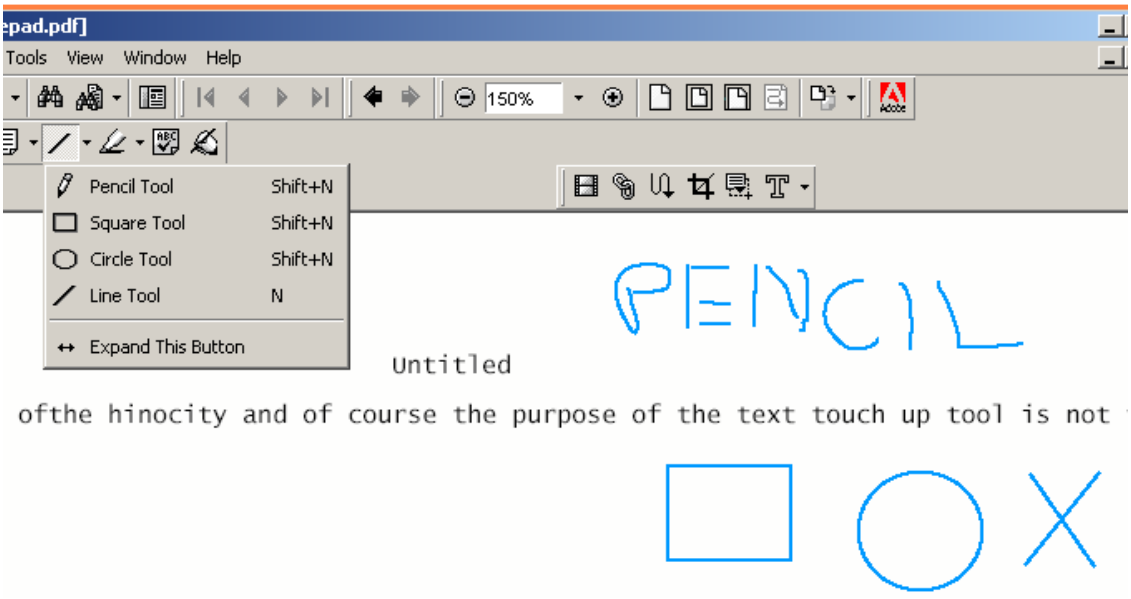


Figure 16. The line tool and its derivatives in action

The line tools are very straightforward (see figure 16). They paint lines of any color upon the canvas. The difference between each of the tools is simply the way that they are created. The pencil tool is done in free hand with the mouse. The square creates rectangles and the circle creates circles. The line tool creates lines, which are always more or less straight.

The highlight tool (see figure 17) gives you the ability to modify text in a pre-

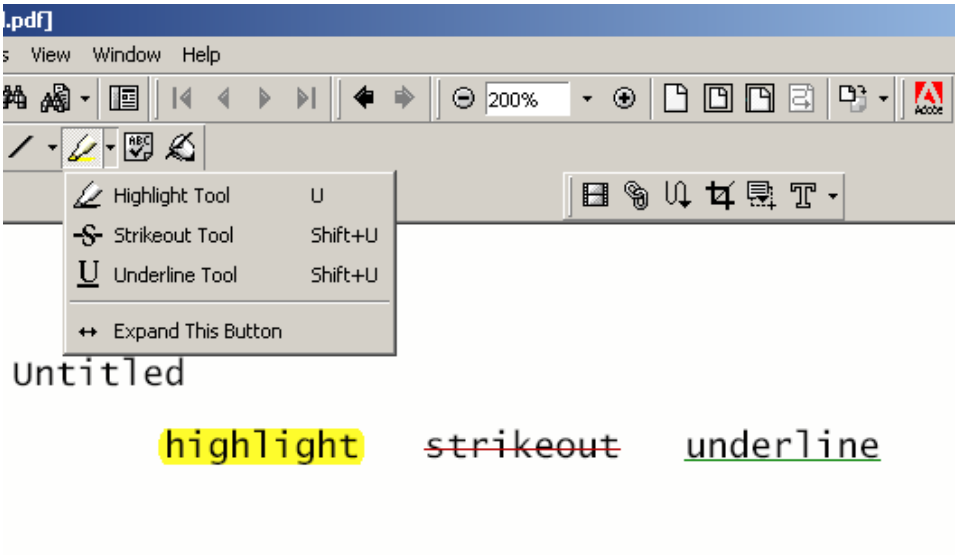


Figure 17. The highlight tool and its derivatives

formatted manner. The highlight tool highlights text with any color. The strikeout and underline tool follow the same functionality, obvious differences aside.

1.2.3.4. Navigation Tools

One of the particularly exciting elements of PDF is its ease of navigation, but finding information easily does not happen by accident. Every object in the file has some sort of addressing associated with it and there are specific ways to use that addressing.

Even the author and editors are things that follow the PDF file everywhere it goes. A handy way to use this information is to create a user account with a document so that you can create digital signature objects. The digital signature requires that the user or editor create an account that contains the information to be encrypted in the signature (see figure 18). The account resides as encrypted information in the page description portion of the file model. The account contains information on what the user creates and allows them to create security policies on them. While security policy is not covered in the scope of this manual, it is still a good idea to create a password for your account so that the information about you is secure.

Besides creating signatures, a user account allows you to create security certificates that are recognized by web-browsers and Acrobat applications. This helps to ensure that your users know if the document has been viewed or tampered with before they receive it.

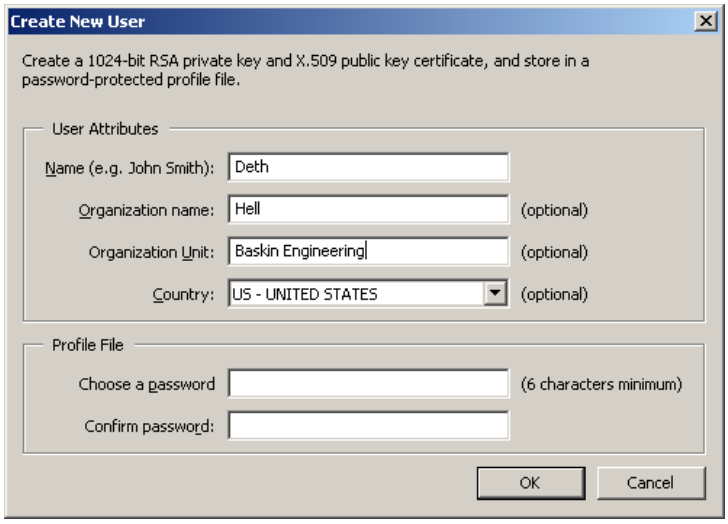


Figure 18. The account setup window will automatically appear when you create your first signature object.

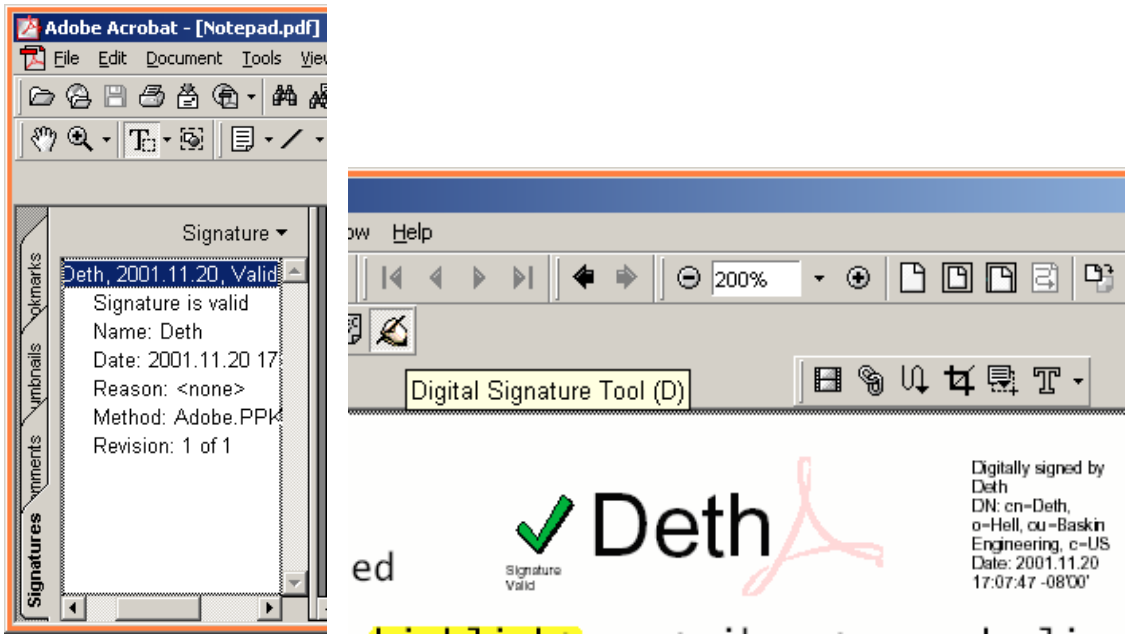


Figure 19. The digital signature navigation tab and the signature itself

The digital signature is an object that is created from your user account and generates a stylized graphical (not graphics) object that displays your information (as in figure 19). If the text of a file is manipulated after the signature has been created, then the green check mark will change into a red "x" to alert the next viewer or even yourself that something has changed since you placed the signature.

The window menu can open the navigation tabs by selecting the Bookmarks, Thumbnails, Comments, or Signatures option from that Menu. These options will only appear in the menu if these objects are present in the file. Bookmarks are created by default but can be changed as need to create chapters or sections to your document. The signatures tab allows you to look at the signatures individually by selecting them in the navigation window and also locates the signatures objects you are interested in viewing.

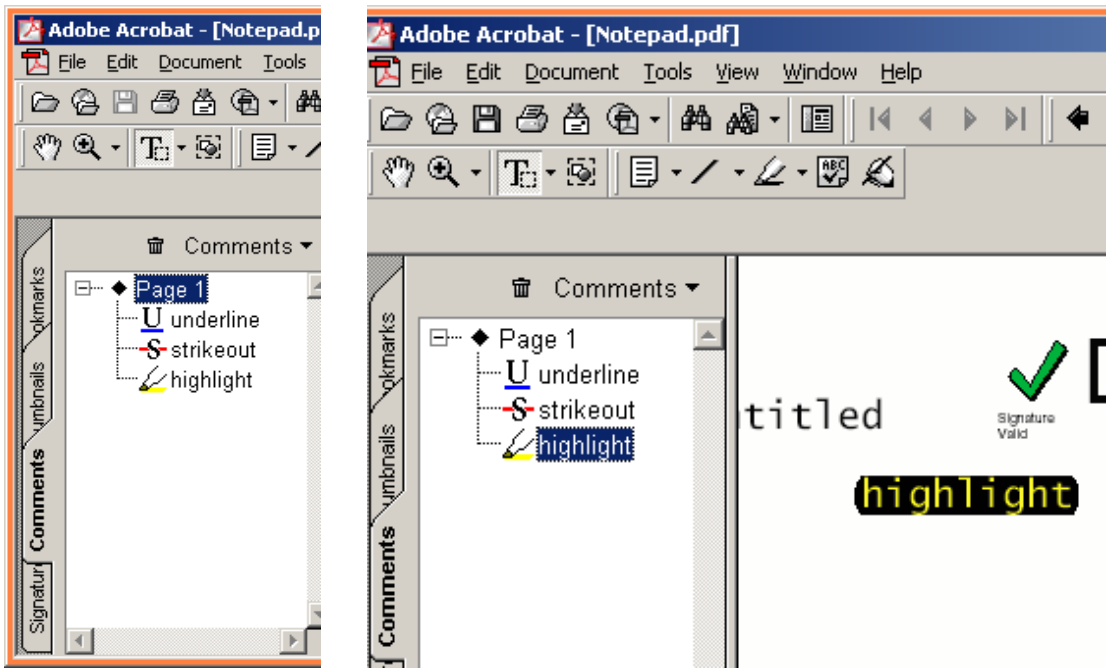


Figure 20. Highlight objects also have navigation information associated with them.

Highlight objects are usually used to indicate something important or that needs editing. As such it has its own tab for locating highlight objects (see figure 20).

The thumbnails tab (see figure 21) is created by default and contains thumbnails pictures of each page object in the file. These are objects themselves and can be manipulated independently but are very handy for navigational purposes in general.

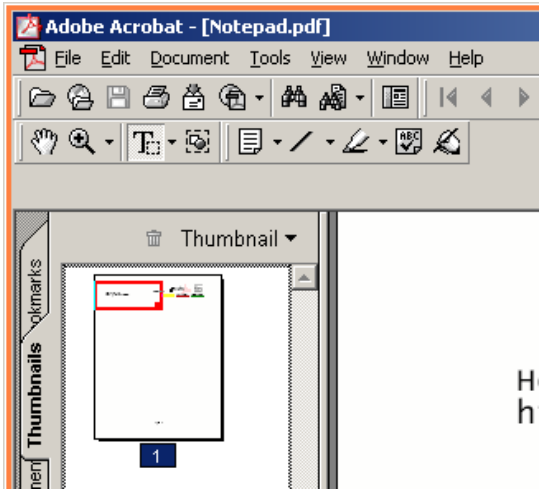


Figure 21. The thumbnail's tab allows you to see a "snapshot" of every page

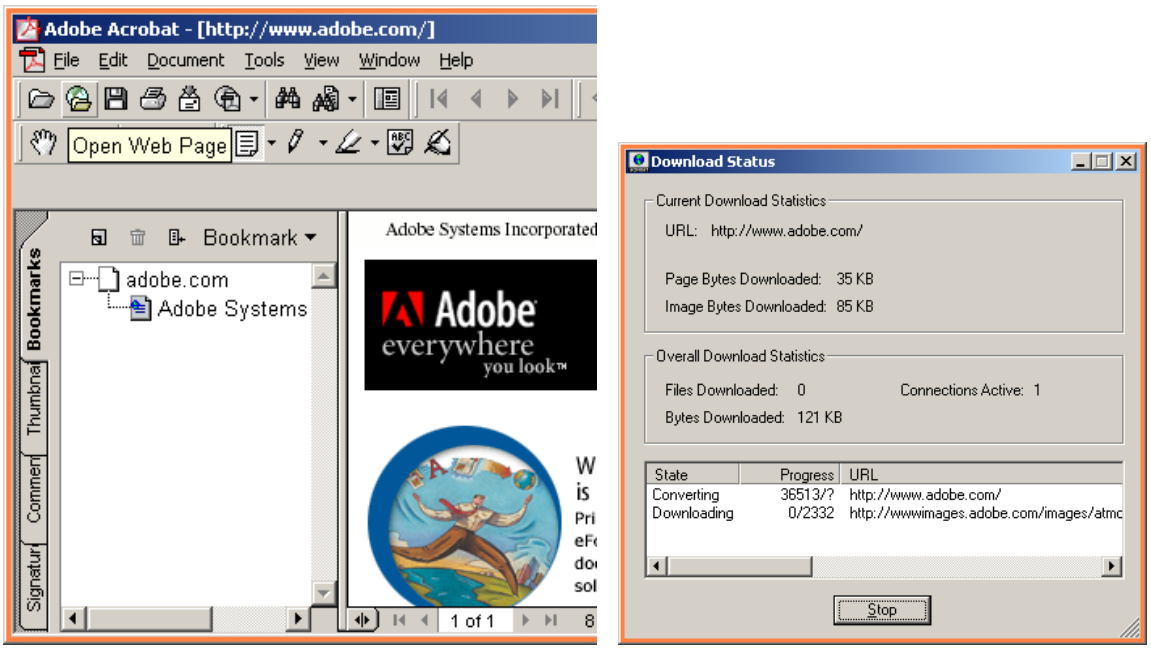


Figure 22. Content in web format can be directly changed into PDF files with the standard edition

1.2.3.5. Web Tools

It has been said that PDF is a good web format. With its small size and attractively priced context application (Acrobat Reader is free), it is little wonder that it is used to create web content. It contains, at the very least, a plug in for Internet Explorer and Netscape that make browsing Internet PDF files easy.

Incidentally, Acrobat can create PDF out of a web page as well. Until recently, that was limited to HTML, JavaScript, Java, and ASP. The 5.0 versions now integrate XML/XSL and JSP as well. The Acrobat standard edition can create a PDF document out of a web page when you click on the Open Web Page (see figure 22) button on the tool bar. It can then be saved and manipulate like any other PDF document. Direct applications for this feature are limited but it can be a formidable tool when it is needed.

1.3. Appendix 1: Implementation Limits

Like any program, Acrobat must manage its own memory resources and architectural requirements. Memory resources define how many memory objects it can manage at the same time. The program itself has but one architectural limit; it can manage a file of not greater than 10 GB exactly. This is because it uses 10 decimal digits to address byte memory. There are also memory considerations but they are merely provided as a minimum memory

requirement for the application, 32 MB, because this is a hardware specific performance issue.

Architectural limits are not merely the amount of space an application can manage but also the size of the variables that it can manipulate. For an integer variable, it can use the full 32-bit range, or the largest positive and negative numbers that can be represented by 32 bits or (2^{31}-1) to (–2^{31}) values of an integer.

The real (or float as they are commonly called) numbers also have a full 32-bit range but are limited to a 5 decimal digit range of precision. And though most of these variable types are transparent to the user, there may be no more than 2^{13} number of array structures, 2^{12} dictionary objects, 2^{16} characters in a string, and 2^{8} characters in a variable name. It is more important for the general user to know the following restrictions:

They are limited to 106x106 pixels (approx. 1/8 scale or smaller for an 8.5x11 image) when making thumbnail objects

The page is limited to 0.25x0.25 inches at its smallest and 200x200 inches at its largest

What about Linux? Well, since the invention of postscript, the Open Source community has worked diligently to acquire the specifications necessary to use it. Because it is a programming language, it was not long before third party and freeware applications started appearing to handle this interpreted language. Adobe has not been as free with PDF. It remains a mostly compiled format that does not lend itself well to "decompiling" without knowledge of the source code of the Acrobat software.

But that is not to say they haven't tried. Though less widespread than ghostscript, the xpdf application 0.91 can, in most cases, accurately display and edit a PDF file. Because of their lack of internal knowledge of Adobe software methodology, this program uses decryption to extract data and allow it to be manipulated. Because this software effectively allows anyone who can download it to crack the security of a PDF file, all of the open source authors of this software are required to report their URLs to the FBI. This may come as a shock to some readers but this is the standard procedure for open source software that allows private citizens access to protected data. This is by no means a discouragement or encouragement to pursue a particular PDF interpreter but instead merely a description of one's options. It is yet to be seen if Adobe can maintain control of this flourishing technology.

1.4. References

1. Bienz, Tim, Cohn, Richard, and Meehan, Jim; *PDF 1.3 Reference Manual*, March11, 1999, PDFSEPC13.pdf from www.Adobe.com, pg. 12-517

2. Wiengartner, Peter; *A First Guide to PostScript;* Dec. 21, 1996, www.cs.indiana.edu/docproject/programming/postscript/postscript.html

3. Bender, Emily, Sag, Ivan A., and Winslow, Thomas; *Instructors Manual for Syntactic Theory: A Formal Introduction*, hpsg.stanford.edu/book

4. Sun Microsystems; *PDF and PS Versions of JDK 1.1 Specifications*; Aug 23, 2001, Java.sun.com/products/jdk/1.1/dowload-pdf-ps.html

5. Smiers, Leo, Nugent, Tony, and von Leitner, Felix; *Xpdf: Decryption;* Oct. 25, 2001, www.foolabs.com/xpdf/decryption.html

6. Shan, Jerry Wei-Jen; *Mixing and Isosurface Geometry in Turbulent Transverse Jets;* California Institute of Technology, 2001

7. The Internal Revenue Service; *About PDF Formatted Files;* Sept. 20, 2000 www.irs.gov/prod/forms_pubs/pdf_info.html

8. Adobe Systems Incorporated; *Adobe Solutions Network: Developer Program*; 2000-2001 partners.adobe.com/asn/developer/main.html

9. Adobe Systems Incorporated; *Overview and Installation Instructions for Macintosh PPD Files,* October 23, 2001 http://www.adobe.com/products/printerdrivers/installmacppd.html

10. Duebert, John; The Acumen Journal, April 2001, *Making PostScript for PDF;* 2001, Acumen Training