

Chapter 14

Introducing Acrobat

by Aandi Inston

In this Chapter

What is PDF?

Viewing PDF

Making PDF

Editing PDF

Optimizing PDF



What is PDF? What is Acrobat? You may already have encountered one or the other. Very likely you're not sure which is which! In this chapter I'll take you on a tour of the world of the PDF format and the Acrobat toolkit, and how they fit into today's Web.

Acrobat isn't new. Adobe estimate that over 10 million copies of their Acrobat Reader have already been distributed. If you've only used Acrobat versions before 3.0, now is the time to look again, as Adobe have worked hard to make Acrobat 3.0 a first class tool for Web publishing.

What Is PDF?

Just what the world needed—another file format! That's all PDF really is. You might think that the world already had enough file formats, but in this chapter, I explain why PDF came about and why the world needs it after all.

PDF stands for portable document format, and portability is the key behind the design of PDF. In a complicated world, PDF is designed to make things simpler by being more portable.

To understand how and why PDF was invented, you have to forget about the Internet and look back a few years at corporate America.

The Problems of Traditional File Formats

Shipping time is fast approaching! Marcia in marketing has a great proposal. She used Microsoft Word to put it together with a great layout and included some embedded charts from Excel. The company president wrote a memorandum with his favorite word processor, Ami Pro. Meanwhile, Peter in promotions designed a new brochure using PageMaker on his Macintosh, and Terri in technical writing prepared the latest product manual in FrameMaker on UNIX. Everyone is really happy with their work and needs to send it to the others on the list. Because they don't work in the same city, they really want to use e-mail as the method of transfer.

How do they send their documents of various formats to each other via e-mail? In the past, these employees had three choices:

- One choice involved making sure everyone has a system capable of handling every document that is produced. This makes the software makers happy, at least, because everyone needs a copy of all of the software in use company wide. As a result, the financial people produce a very short list of authorized software.

Almost immediately, fonts turn out to be a problem. Marcia used her favorite fonts, but the president doesn't have them on his PC. Word substitutes without a problem, but suddenly, the document has different line and page breaks, and the look is spoiled. The wise people in charge produce a list of approved fonts, too.

This approach is actually quite widespread. A typical solution is to use Microsoft Word for everything, which seems great if you aren't involved in designing or writing books. It's not that great if you just mastered your last word processor or if you're on a UNIX system. This battle is constantly re-enacted across America and the world, and nobody has heard the last of it.

- Reducing everything to the lowest common denominator is an option, which usually means choosing plain text (ASCII) format. Marcia loses her charts and layout; the president loses his letterhead (no big deal, perhaps, but guess who makes the most noise?). Terri loses all the illustrations in the book, as well as producing an ASCII file that is too large for anyone to read. Peter is particularly unhappy because all his design is reduced to a few dozen words of copy.
- The other choice is even more widespread. Print the document and ship everyone a copy. Printer sales are booming; color printers are becoming the norm in every office. Why worry? Well, all this printing costs a great deal. It adds delays, especially if the paper has to travel to another office. Above all, it upsets the people who were told computers would lead to the paperless office.

What if everything could be converted to a universal format? Such a format is often suggested by people who don't really understand just how different are the needs of the many different applications. However, Adobe was among the groups wondering if there really were a solution.

In solving the problem, they quite accidentally invented something that would be ideal for solving many of the problems of Web delivery, a few years later.

PDF: Electronic Paper

All the applications that people use to make documents have very little in common, but one thing they can all do is print! Adobe's idea was to use the print function to create a new document in a universal format. Acrobat pretends to be a printer, but instead of printing, it produces a PDF file.

To succeed, PDF had to satisfy several key requirements:

- *Quality*—the documents must look as good when they are printed or viewed on screen as the originals did.
- *Accuracy*—the pages must look just as the designer intended. Text must not *reflow*: move about to fit in a different space just because the recipient of the document has different fonts. (It's no big deal if the president's memo reflows, but if Peter's brochure text reflows, the advertising copy might not fit in the spaces assigned to it.)
- *Completeness*—you shouldn't need to remember to include extra files with the PDF file.
- *Compactness*—PDF files must be as small as reasonably possible, especially if they are to be e-mailed.

You could satisfy some of these requirements by simply creating a bitmap of the pages, such as a TIFF file. However, almost nobody uses multi-page TIFF files as a universal format. If they provided the full solution, PDF would be no big deal. Bitmap images are accurate and complete, but they can be very large, especially if they are large enough to maintain enough of the quality of the original to print small text clearly, for example. These large files also display slowly.

PDF is a hybrid format. It has bitmaps where it needs them—of course, it needs them if the original file had photographs. It can include real text that can be displayed at the maximum quality supported by the screen or printer. PDF can also include drawn graphics, which is ideal for charts and boxes.

Acrobat: a PDF Toolbox

Acrobat and PDF—which is which? PDF is the file format, and Acrobat is the suite of applications that Adobe produced for working with it. A veritable toolbox, Acrobat consists of several different applications:

- *Acrobat Reader* can read and print PDF files. It is now free of charge, but it's easy to forget that it wasn't always free. In fact it was originally available only in packs of 50 or more! If this sounds bizarre, think back to the original goal of achieving the paperless office. One of the characteristics of PDF is that people keep finding new uses for it, and Adobe's decision to make the Reader free has helped this evolution.
- *Acrobat PDFWriter* replaces the printer driver in Windows or Macintosh to produce PDF files instead of printing.
- *Acrobat Exchange* works like Acrobat Reader and gives some simple editing options, such as deleting or rotating pages or merging files. Acrobat Exchange also adds links and bookmarks.
- *Acrobat Distiller* converts PostScript to PDF. It's a more powerful and flexible alternative to PDFWriter, but it's not as easy to use.
- *Acrobat Capture* is a newcomer to the Acrobat stable. It allows scans and TIFF files to be converted to PDF and attempts to recognize the text in scans to convert it to real text.
- *Acrobat Search* and *Acrobat Catalog* let you create indexes to PDF files and search them (such as finding all the PDF files on the CD-ROM containing the word "unleashed").

Adobe has repackaged this combination of products in countless different ways, changing at each release and sometimes between releases. Acrobat Pro 2.1 included all the preceding utilities except Capture and Catalog and then added Catalog midway through the life of the product. Acrobat 3.0 contains almost everything.

The only thing left out of the Acrobat 3.0 product is the full Acrobat Capture product. There is a Capture plug-in for Exchange, which is all that most people will need, but the full product is more suitable for heavy use.

WHAT AROBAT ISN'T

Acrobat is a much misunderstood product. Given the basic fact that Acrobat can convert any document to a portable, sharable, form, many people lost no time in jumping to all sorts of conclusions about what it does.

The most common conclusion is that Acrobat is some sort of universal file converter. In a sense, of course, it is, but people ask, "Will Acrobat help me read Macintosh Quark Xpress files on my PC?" Yes, of course, but only if the creator uses Acrobat tools and makes a PDF file for you. Another example of this assumption is a question such as "I can't get Acrobat Reader to open GIF files."

The next misapprehension is that Acrobat is some kind of word processor or desktop publishing application. People ask "How do I create my documents in Acrobat?" The point is that you continue using your favorite applications.

Many people assume that PDF files are just like word processor files. They have a PDF file—perhaps they made it themselves and threw away the original or maybe someone else sent it to them. Now they want to edit it. As you learn later in this chapter and in chapter 16, "Making PDF", Acrobat offers some limited options for editing, but none are half as good as returning to the original document.

The most unusual expectation of Acrobat I've ever heard was from someone who wondered if it would improve the quality of output on his PC when it was connected to a mainframe as a terminal. I still haven't figured out how that idea came about!

Viewing PDF

To view a PDF file, you need a PDF viewer. It's no surprise that Acrobat Reader is the most common PDF viewer. You can also use Acrobat Exchange to view files, but it works identically, so I'll only discuss the Reader.

Acrobat Reader is free and available for most common computing platforms. There are versions for both Macintosh and Windows. Acrobat 3.0 offers a version optimized for Windows 95 and Windows NT, as well as a separate version for Windows 3.1. There are also versions for OS/2 and many popular versions of UNIX.

Increasingly, people will already have a copy of Acrobat Reader on their computer (for instance, it ships with new Macintoshes), but if you are producing PDF, you should at least tell people how to get the Reader. On the Web, this is particularly easy—you simply send people to Adobe's Web site. It isn't expected that you provide a copy to download independently (although there is nothing to stop you).

Adobe's Web site is at <http://www.adobe.com>. Adobe recommends that links to get the Reader point directly to <http://www.adobe.com/acrobat/readstep.html>, and even provide an icon you can use, though you may have to complete some paperwork to be allowed to use it.

You might already have a copy of the Reader. It is included with all the Acrobat products and comes on the CD-ROM included with this book.

You might already have a copy of the Reader. It is included with all the Acrobat products and comes on the CD-ROM included with this book.

If you're running Acrobat Reader 2.1, I recommend that you upgrade to 3.0 soon for two important reasons:

- The version 3.0 Acrobat tools can produce files that earlier versions of the Reader cannot read. Because these files can be 20 percent smaller, the new format will be popular.
- Acrobat Reader 3.0 integrates with the leading Web browsers to let you view PDF files within the browser screen, rather than in a separate window. To some people, this might sound like a gimmick, but it's an important part of making PDF a first-class member of the Web community. It can mean that the Web user views HTML pages and PDF pages in exactly the same way, and this consistency of design is very important to smooth navigation.

Figure 14.1 shows a document viewed in Acrobat Reader, and Figure 14.2 shows the same document viewed in Netscape 3.0. Behind the scenes, Netscape has asked Acrobat Reader 3.0 to do the hard work of showing the PDF file—Netscape doesn't really understand what to do with PDF files, unless it gets some help!

Navigation—the Key to Interactivity

Everyone who's used the Web or a CD-ROM-based product knows that a good site or product offers more than just eye-catching content. Above all, you need navigation. The two kinds of navigation I discuss are traditional PDF navigation (with the PDF files all on your hard disk or a CD-ROM) and PDF navigation for the Web (with the PDF files being downloaded by your browser). The

- In any document with multiple pages, the user can move between pages by clicking navigation buttons, using a scrollbar, or typing in the page number. The creator of the document doesn't have to do anything to make this to happen.

This is rather different to a Web browser, where an HTML page is one long piece that you can just scroll up and down; with PDF the document is divided into separate pages, just as if it was on paper. Like a Web browser, the Reader offers a Back button to step back through the sequence of pages visited.

- Documents can contain *links*, which are areas of the page that lead to another page when clicked on. These are quite similar to the links in an HTML page to take you to a different part of the same page. Links can include graphics or colored, underlined text.

It's important to realize that the link is actually just a rectangular area of the page that contains the text or image. If you create a link on some black text, it doesn't change color or get underlined, although you can draw a box around the link. This is different from HTML, where making a link from some text automatically changes its appearance.

- You can set up links to move to a page in a different PDF file. The Reader automatically finds and opens the file and then moves to the page. Again, this is similar to links between HTML pages. You can even set up links to run a different program or open a non-PDF document, although once you do that, you must provide a way to navigate back to the Reader.
- Documents can contain *bookmarks*. Bookmark isn't perhaps the best term for this item, which is really a form of table of contents. Bookmarks appear in a window to the left of the page, and you can set them up so that you click the chapter titles to open the chapters. Click on the bookmark itself to go to the page.
- You can put *thumbnails* in your documents in place of the bookmarks. Thumbnails are miniature pictures of each page, so this choice is suitable if each page is very graphical and obviously different from its neighbors. Users can move to a

page by clicking on its thumbnail. You can actually use both thumbnails and bookmarks, but only one of them can be seen at a time.

PDF on the Web offers the following navigation tools:

- PDF documents on the Web have pages too. Although the traditional Web document is an HTML page that is infinitely long, PDF documents are divided into separate pages just like printed documents. You can scroll within the pages of a PDF file on the Web.
- Links inside a Web PDF file, linking to a different page of the same document, work just the same on the Web as when used with traditional PDF.
- Links to other PDF files have to be handled rather differently. When the link is made, Acrobat stores the filename of the other PDF file. In Acrobat 2.1, when you click the link, the Reader looks for a file on your hard disk, not on the Web. In Acrobat 3.0, Acrobat Reader is smart enough to know that if a document came from the Web, links in that document are on the Web, too. This allows you to move a collection of PDF files from your local disk onto the Web, without having to remake any of the links.
- Acrobat 2.1 added a new kind of link. A *Web link* is a link to an object on the Web. Typically, it's an [http:](#) reference to another Web PDF file, but the file doesn't have to be in PDF format. Originally, Web links were rather clumsy; Acrobat asked your browser to follow the link, and if the link was a PDF file, the browser handed the PDF file back to the Reader.

In Acrobat 3.0, they are much friendlier, because the PDF files are all shown in the browser window. As links to PDF files in the same place are followed to the Web anyway, you need only use a Web link to point to a different site.

Making PDF

The key to making PDF is the print model. Any successful appli-

cation used in page layout or design must be able to print what the user expects. Acrobat hooks in at this stage to make a PDF file instead.

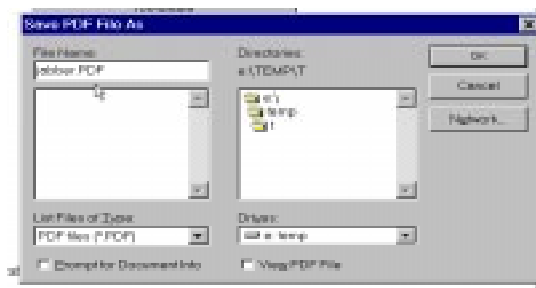
Chapter 16, "Making PDF," covers every aspect of creating a PDF file in more detail, but I'll outline the basics of the two Adobe tools: PDFWriter and Distiller. I'll also discuss some of the other ways to make PDF files.

PDFWriter and Distiller

PDFWriter is easier to use than Distiller. It is suitable for many applications except some of the high-end ones. You create a PDF file in PDFWriter with the following steps:

1. Select the PDFWriter printer driver. On a Macintosh, you do this using the Chooser; in Windows, you use the Printer Setup item from the File menu.
2. Select Print. Choose any appropriate options, such as Print All Pages.
3. You are prompted for a name for the PDF file. PDFWriter attempts to suggest a name that starts with the current document name and ends in .pdf. Figure 14.3 shows PDFWriter's prompt.
4. Switch back to your original printer driver if you want to print a hard copy of the document as well.

*14.3.
PDFWriter prompting
for a name for your
PDF file.*



That's it—you now have a PDF file. Word and Excel macros supplied with Acrobat 3.0 make the job even easier for someone using those tools in Windows 95. The Create Adobe PDF option appears on the File menu to combine all the preceding steps except the file-

name prompt.

With Distiller, creating a PDF file is a bit more complex:

1. Select a PostScript printer driver. You need to make sure you can print to a file, so in Windows, you might need to set up a separate driver if your application doesn't have a Print To File option.
2. Choose Print. You are usually prompted for a name for the PostScript file.
3. Once the printing is done, switch back to your original printer driver if you need to.
4. Start the Distiller application if you haven't already.
5. Use Open in Distiller to open the PostScript file you just wrote.
6. You are prompted to name the PDF file. By default, Distiller offers the name from the PostScript file and with .pdf on the end.

Why bother with Distiller? I'll discuss some of the reasons to use Distiller instead of PDFWriter.

The most important reason is that some applications just don't work very well when they aren't dealing with PostScript. Quark Xpress is the best known example of an application that really doesn't print well to non-PostScript printers.

The second reason is that Distiller provides extra options that are not available in PDFWriter. For instance, Distiller is better than PDFWriter at reducing the resolution of images to suit screen display, making smaller PDF files.

It's sometimes easy to forget, but some applications in the world, such as DOS and UNIX applications, still don't run in Windows or on the Macintosh. They can't use PDFWriter, but most of them can write PostScript (via their own printer drivers). Then, you can distill the PostScript. Distiller is also available for most popular UNIX systems.

Finally, Distiller allows you to insert extra instructions (called *pdf-marks*) in PostScript code. These instructions do nothing on a

printer, but when distilled, they can create extra PDF features such as links and bookmarks. Some applications, such as FrameMaker, can write pdfmarks automatically, making the creation of links much simpler.

Why Not PostScript?

Some people ask, "Why invent PDF when PostScript was already there?" Someone equipped with a suitable tool could view and print PostScript files in much the same way as the Acrobat Reader handles PDF—so why bother with PDF?

The question is a good one, and it's worth remembering that Adobe owns both formats. They must have felt that their own baby, PostScript, was not up to the job. Here are some of the reasons:

- PostScript is limited in portability. Not everyone believes this, and some people continue to push PostScript as a standard for the Web, but PostScript usually includes instructions specific to each printer, which can cause problems if transported. The classic example is page size—a document created in the U.S. probably uses letter-size paper (8 by 10 inches). In Europe, the standard size is a4. Many documents that request letter-size paper stubbornly refuse to print if the printer only has a4!
- PostScript is usually much larger than the corresponding PDF. Although it is possible to have PostScript that is compressed, few drivers write it that way.
- PostScript doesn't have any way to handle font substitution. All a printer does is substitute the ugly Courier font if the exact font required is not available.
- PostScript is difficult to access at random—pulling page 700 straight from a 1000-page document, for example—which PDF can handle quickly because it has an index.
- PostScript has no provision for links and other interactivity.
- PostScript is a much more complex language than PDF, and it is usually slower to display. PDF was designed so that the program which created the PDF file did most of the hard work before creating the PDF. PostScript allows enormous flexibility, but at a cost.

Given all these reasons, Adobe decided to invent a new format, but they didn't start from scratch. PDF salvages a good many of the successful ideas from PostScript. Adobe did a similar job of making a simple PostScript when they designed Adobe Illustrator, which needed a file format but couldn't handle full-blown PostScript. They will do it again with the forthcoming Bravo—see Chapter 2, "Adobe Fonts Online". Why waste a good idea?

Ironically, Adobe has announced that a PostScript level 3 will be available later in 1997. This will include all of PDF as an integral part of PostScript, bringing the products back together.

Other Tools: Capture and Things...

Using PDFWriter or Distiller isn't the only way to make PDF files. A more recent addition to the Acrobat family is Acrobat Capture.

Acrobat Capture is another example of an Acrobat product finding a market that Adobe wasn't expecting. Adobe was aiming at a comparatively small market: document archiving. Many large organizations have warehouses full of paper that might need to be referred to. Transferring this paper into electronic form is very attractive to them because it saves storage space, saves employee's time, and makes documents available more quickly. With a bit of luck, companies are also less likely to lose the electronic documents. Imagine the consequences for an insurance company if it misfiled 5 percent of its paperwork—it is as good as lost. This figure is not unrealistic.

The idea of scanning paper and saving it as bitmaps was already well-established, but this option suffers from drawbacks. It occupies more storage space, and the only way to extract information is by pulling up the bitmap and having someone look at it. There's no easy way to search for documents except by applying a filing system on top of them.

Optical character recognition (OCR) is the solution to the drawbacks of bitmaps. OCR is still an inexact science; as with speech recognition, it is pretty hard to reliably understand the contents of a printed page. Still, Capture offers a help here—if it can't recognize part of the document, it keeps it as a picture. Nothing is lost (except the capability to search on the words that can't be recognized).

Capture was originally a very expensive tool—several thousand dollars—which reflects the market that Adobe expected. A succession of price cuts led the way for Adobe to include a Capture plug-in with the Acrobat 3.0 product, although the full Capture product has features intended to help with the management of large volumes of documents.

The Capture plug-in is ideal for transferring small numbers of existing paper documents to the Web. And it's built right into Acrobat Exchange, so you can scan, import bitmap files, and perform OCR, all without leaving Exchange.

Adobe PageMaker has a Create Adobe PDF function, which is

really just a convenient way to create a PostScript file and run Distiller. It does also give an opportunity to automatically create pdf-mark instructions for bookmarks from a table of contents, and for links from an index. PageMaker 6.5 lets you create additional links that work when the document is converted to PDF or to HTML.

Adobe Persuasion 4.0 lets you create PDF files too, and create links within the file using simple drag-and-drop.

Not to miss out, Adobe Photoshop 4.0 can create PDF files. This is most suitable for a graphically intensive page with no text or a very small amount of decorative text. Bitmaps of text are not recommended for PDF files where a body of text has to be read, as the quality will be poorer than text made in other ways.

Editing PDF

I already emphasized that PDF isn't a word processor format and that it's important to keep the originals of any file. It's almost always easier to change that original and make a new PDF file than to change the PDF file. So what is this section about?

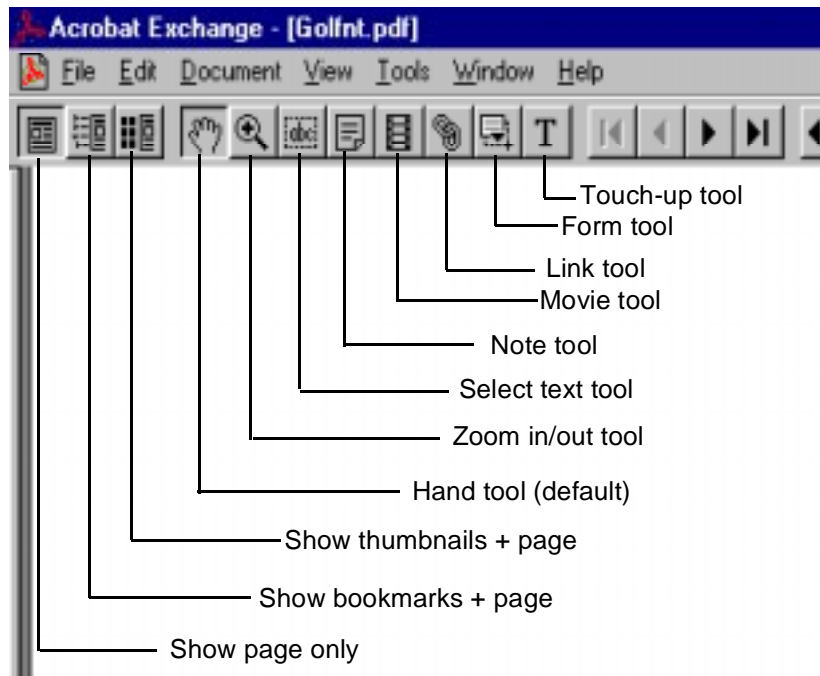
The main tool for editing PDF files is Acrobat Exchange. This section outlines what you can do with Exchange.

Exchange lets you edit PDF files but not change their contents. What? Actually, Exchange deals with pages. Exchange lets you perform the following tasks:

- Delete pages
- Combine the pages from two PDF files into one
- Crop pages (remove white space around pages that are too large)
- Rotate pages (which is important because a lot of the PostScript to print landscape pages actually turns the pages on their side)

Figure 14.4 shows the buttons of the toolbar in Acrobat Exchange, together with their functions. There are many more functions that can only be reached from Exchange's menus.

14.4.
The Acrobat
Exchange toolbar



Touch-Up Plug-In with Acrobat 3.0

I've been going round for years telling people that you can't change the contents of pages with Exchange. Now Adobe have proved me wrong by introducing the Touch-up plug-in, which is a part of Exchange.

The Touch-up plug-in can be used to change existing text on a PDF file. But don't get carried away! It is designed for correcting very small mistakes within a line of text.

Touch-up is rather like correcting the text on a piece of paper, using an eraser. You can replace text, but you can't move lines around to make extra text fit. You can't even use Touch-up to add text where there wasn't any to start with.

The real purpose of Touch-up is to correct errors introduced by the Capture plug-in, because optical character recognition (OCR) is rarely perfect. You may find it handy to correct small typing errors,

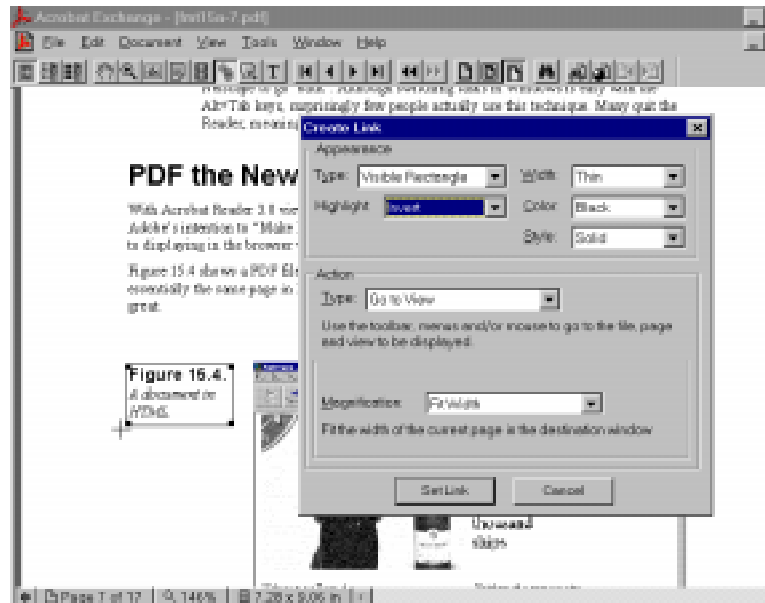
or to change dates. If you insert a page into the middle of a document, it may be reasonable to use Touch-up to change the printed numbers on the following pages, if there are not too many.

Links

Links are what transform a document that is just paper on screen into an interactive experience. As described previously, there are different kind of links, bookmarks, and so forth. Adobe call these collectively *annotations*.

Creating a link with Exchange is very simple. A link has a starting page and a finishing page. You click the starting page, and the link takes you to the finishing page. In Exchange, you create a link with the following procedure:

1. Move to the starting page.
2. Select the Link tool—this is an icon of a paper clip.
3. Drag a box to outline the link. For instance, drag a box around a phrase you want to link from.
4. A Create Link dialog box appears, as shown in Figure 14.5. Don't use it yet!



5. Move to the finishing page.
6. Now, click Set Link in the Create Link dialog box. The link is set, and Exchange returns to the starting page.

This task seems simple in itself, but adding a number of links to a large document can be tedious.

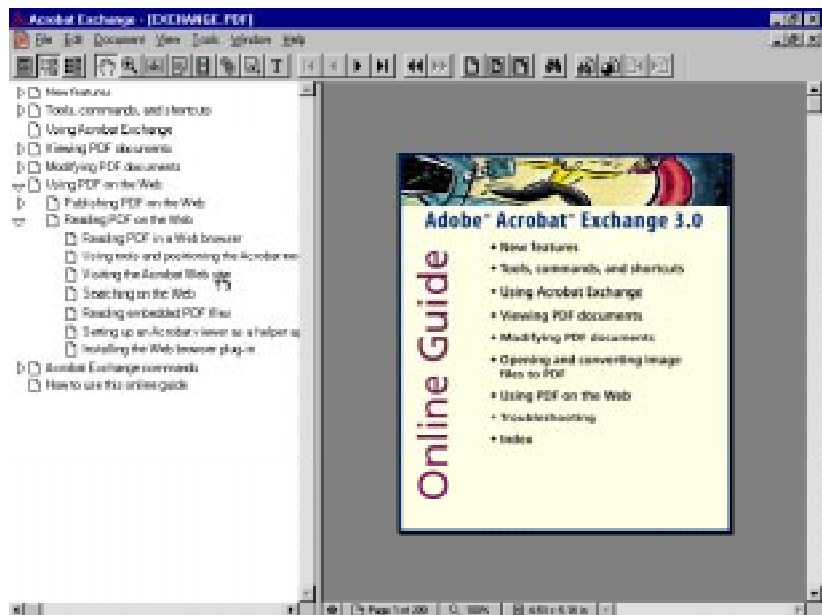
Other Kinds of Annotations

Exchange can add other annotations. In fact the list of annotations is open-ended—by writing a plug-in, programmers can create new types of annotation. This section describes some of the main annotations and how they are used.

Bookmarks appear in a window at the left of the screen after you click the Bookmarks button (see Figure 14.6). The Bookmarks button is the second button in the toolbar. The procedure for creating bookmarks is a little simpler than the procedure for creating links because there is no starting page to worry about. Just move to the finishing page and use Ctrl+B (Windows) or Command+B (Macintosh). You now have a bookmark pointing to the current page, and you can type the title for the bookmark.

T I P

If you get hold of a great new plug-in that offers fancy annotations, think carefully before using it in a document for use on the Web. Will the person downloading the document have that plug-in? If not, what difference will it make to the look of the document? It's useful to have an unchanged copy of Acrobat Reader somewhere for testing.



14.6.
*Acrobat Exchange,
showing bookmarks.*

T I P

When you create a bookmark, select some text in your document first. This text then becomes the text of the bookmark, saving time. Before you can select text, you must click on the Select Text button.

Bookmarks can be nested like folders. Top-level bookmarks can be opened or closed by clicking on the triangle icon. To create bookmarks that work in this way, click a bookmark's icon and drag it to the left or right.

Thumbnails appear instead of bookmarks if you click the View Thumbnails button (see Figure 14.4). You create them using Document | Create All Thumbnails item from the Exchange menu.

You create *notes* using the note tool, after clicking the Note button (see Figure 14.4). Drag the mouse over the page to create a window to type notes. Originally intended for a group of people to review documents and add their comments, the notes option might be a useful way to send small messages to your readers. Notes can be left open, or closed to a small icon.

Movie annotations are created using the Movie tool (see Figure 14.4 for the appearance of the Movie Button). Drag a space for the movie screen and then choose a QuickTime (Macintosh or Windows) or Video for Windows *.avi* file (Windows only). The movie stays in a separate file, rather than becoming part of the PDF file.

Article threads are under-used—quite possibly because they don't have a button! Use Article from the Tools menu to define them. Article threads allow you to guide the reader through a piece of text that is too long to fit on the screen all at once. They can easily follow a story through columns and continue to different pages. The user can also view a list of the articles you created with the titles you gave them. I'll discuss articles some more later, in Chapter 16, "Making PDF".

Introducing PDF forms

Acrobat 3.0 introduced a new feature to PDF—forms. This is currently only available for the Windows and Macintosh versions of Acrobat Reader, so everything to do with forms is invisible in any other version of the Reader.

When someone says forms to me, my eyes start to glaze over. Filling in forms is, to me, one of the least interesting activities in the whole world, and designing them hardly seems more interesting. But when Adobe decided to add form-filling to Acrobat, they added so many features that it has opened some really exciting possibilities for PDF. In fact, forms can be used to do many of the things that

JavaScript can do for HTML, without a line of programming!

Sure, you can use PDF forms to add a place to type name, address and social security number. And you can add pull down lists for State, or buttons to choose married/single. You can use PDF forms to take an existing paper form and allow it to be filled in online. The forms can be submitted to a Web server, just like an HTML form. One use of this would be to design attractive Web Search pages.

The basic elements of forms, especially buttons, can be used to do much more. Here's how:

- Buttons can contain text, but also icons. An icon is a picture taken from a different PDF file, and can be any size. Although icons are usually drawings, they can contain text or bitmaps just like any other PDF file.
- Buttons need not have a border round them. So now you can have what appears to be an ordinary picture, but is actually a form button.
- Form buttons can have a list of actions to be taken when they are pressed. These can include following links, giving you the ability to add pictures which form links. Adobe supply several PDF files of clip art for use as buttons: a mixture of arrows, icons and fun stuff, but you can use your own PDF files as raw material too.
- The actions taken when a form is pressed can also include hiding or showing other form elements. So as you press a button, the button can disappear and/or other buttons (hence pictures) can appear.
- A button can be set up so it is 'pressed' simply by moving the mouse over it.

Taken together, this opens many possibilities for interactive PDF files, and this is a theme I return to in chapter 18, "Unleashing PDF on the Web." In chapter 17, I talk about how to create forms.

Other Editing Tools

Adobe Illustrator has been around for a long time as Adobe's flagship drawing package, and it's no surprise that it is the first product

to offer PDF editing, although its capabilities are rather limited.

Using Illustrator 5.5 and above, you can open a PDF file. What actually happens is that a thumbnail is shown for page one of the PDF file, and you can step through to select the page you want to work on. Then, Illustrator opens that page as editable elements.

This editing tool can be useful, but it has limitations. Not every PDF file can be opened successfully, especially those that use fonts in non-standard ways. Everything on the page is graphical, so using Illustrator to make substantial changes to text is tedious to say the least, especially if you need to move text between pages (which, of course, must be worked on separately).

The big problem at the time of writing is that there is no Illustrator 5.5 for Windows.

Optimizing PDF

Creating a PDF is easy. Many designers want to go one step further and optimize the PDF for the Web. Because it isn't a single technique, optimizing a PDF includes a variety of different tasks.

The most common trade-off is speed versus quality. Always remember that on the Web, quality is not everything. A beautifully designed home page that takes a home user 20 minutes to download with his modem before he can start to view it will not win you any customers! I recommend that any Web designer access their own pages via a 14400 baud modem to see how many of their end users will see it.

When designing for paper, it's easy to say that the final appearance is the only thing that counts, and you can stop when it looks right. Until you are familiar with optimizing PDFs, you should consider that a PDF that looks right is a first step to be followed by optimizations to achieve the smallest file you are still happy with.

Reducing Size

Reducing size is a good thing. It reduces the time it takes the file to reach the user and the amount of disk space needed to hold it. It is also more likely to be cached, and that can speed up access dramati-

ically. Not all the techniques for reducing size can be applied to every file. Here are a selection of techniques:

- Work with low-resolution bitmaps. You can start with them or use Distiller to reduce the resolution if you want to keep high-resolution bitmaps for publishing separately. Be ruthless, and check to see what is an acceptable resolution. For color photographs, you might find you can go below the traditional screen resolution of 72 dots per inch (dpi). Scanned text and line art need higher resolutions. You must judge the importance of the quality of a printed result—if printing is important, you might need higher resolutions, although 100 dpi is adequate for photographs on almost all desktop color and black-and-white printers.
- Work with Distiller's compression options. Distiller lets you compress bitmaps in a variety of ways. JPEG is good for photographs, but the quality settings might be too low for some pictures.
- Reduce the number of fonts. (Multiple sizes of the same font are okay, but different styles—italic and bold, count as separate fonts).
- Consider carefully whether to embed some or all of your fonts. Using PDFWriter and Distiller, you have the choice of whether to include your own fonts inside the PDF file. If they aren't embedded, the file will be smaller. However, the Reader will make substitute fonts, and the document may not look the same.
- If you have very complex drawn figures that take a noticeable time to draw on screen, there's a good chance they are using a lot of space in the PDF file too. Try replacing them with a bitmap of the same drawing.

Improving Delivery Time

Improving delivery time to a Web user was once a simple question of making the file smaller. Now there is much more to it.

The most obvious benefit of Acrobat 3.0 is that the PDF file can appear within the browser's window. Probably more important than this is something you can't immediately see by looking at the

screen, though.

It's possible for Acrobat Reader 3.0 to download PDF files a piece at a time and display the result as soon as it arrives. It might seem so obvious and sensible for the Reader to work this way that it's worth pointing out why this needed a change.

A PDF file contains an index at the end, indicating where everything in the file is located. Each page, picture, font, or block of text is found through the index, and each file can be in any order. If you read a PDF file in order, you have to reach the end before you are ready to display any of it.

For Acrobat 3.0, Adobe made some changes to the design of PDF files. The *optimized PDF* format has two important features, designed to improve the actual and perceived speed of download and display:

- Hints at the start mean there's no need to read the index at the end.
- The format dictates a specific order to include the items in a PDF file. Each page is in order, but there are more subtle changes too: If a font is embedded, it appears after everything else in the page. Text appears before graphics, too. The Reader gets the text of the page and shows it using substitute fonts as soon as it is available. Then, it draws the pictures. Then, it gets any embedded fonts and puts the text on the page again, this time as the designer intended.

Deciding whether to embed fonts is no longer a clear issue. Although embedded fonts make the file larger, they don't prevent the user from seeing the text quickly. If the user can start to read the file sooner, then they perceive the file as quicker even if, overall, it takes just as long.

It's important to realize that you must optimize a PDF file before you can gain these advantages. Fortunately, this is easy in Acrobat Exchange 3.0. You can optimize a file as you save it, or you can use Batch Optimize to optimize all the files in a folder or series of folders, bringing all your existing PDF files up-to-date.

Optimizing isn't the whole story. What if your PDF file is 100 pages long, and the user clicks a link to page 95 as soon as he can see the first line? He still has to wait a long time while the browser

reads 94 pages that don't interest him.

The solution to this is a separate process called *byte serving*. This is not something the creator of the PDF file does, although the Reader understands it. Instead, byte serving is initiated by the Web server. Some Web servers support byte serving already; others need a small change. Once the server administrator sets it up, byte serving is available to every user retrieving PDF files from the server.

Designing for Web Delivery

It's easy to take an existing document and turn it into a PDF file. That's a great strength of Acrobat. However, you can improve on a pure translation of an existing document, adding in the value, appeal, or usability of the document. I cover this in more detail in chapter 18, "Unleashing PDF on the Web," but here are some of the ways you can improve a document:

- Intelligent use of links within the document helps the user get around.
- Bookmarks can be even better for many documents because they can be on screen no matter what page the user is viewing. This means the user doesn't have to return to a known point of reference or finish a section to navigate elsewhere.
- As with HTML pages, Web links to other Web documents on the same or different sites add to the interactivity of the Web.
- Article threads help lead a user through articles that don't fit on a single screen or page, especially if they are arranged in columns.
- If you design exclusively for the Web, why be constrained by traditional paper sizes? On a typical user's monitor, it is difficult to read the text on an entire regular-sized page, so why not try smaller pages, and perhaps different shapes too?

PDF on the Web: Making the Decision

Designers usually like PDF, at least compared to HTML, for the following reasons:

- PDF files always look as the designer intended, whereas with

HTML, you can only hint at the appearance you want.

- It's often easier to make and manage a PDF file than an HTML file (no imperfect conversion and no separate graphics files to manage).

Why would anyone hesitate? It's important to consider the other side too, so here are some of the reasons not to use PDF on the Web:

- PDF readers are not universal. Not everyone has Acrobat Reader. With a slow modem, the time and cost of downloading it can be substantial or even impossible.
- Even if downloading the Acrobat Reader were quick and easy for everyone, a substantial body of users find downloading and installing software seriously daunting. You should consider whether such users represent an important part of your audience.
- Another group of users who might be excluded from your page are those inside a corporation that forbids unauthorized software and does not include Acrobat on the allowed list. This group of users could be more influential.
- Some people work on platforms for which there is no Acrobat Reader. Perhaps it's not a commercially important number, but dedicated users of Amigas and UNIX systems such as FreeBSD may be excluded from your page.
- Sometimes HTML is more flexible. For instance, HTML can usually fit inside the width of the user's window, which can aid in the accessibility of the page.
- Few of the Web searching robots that build indexes for search engines like the one at Lycos (<http://www.lycos.com>) can index PDF files, yet. If you rely on search engines for part of your hits, this may be a problem.

Given these reasons, what is the right decision to make? It's not an easy choice.

I've never found a PDF-only site, although it is possible in theory. (There is the technical difficulty that many servers default to a [page index.html](#)). Personally, I favor sites that offer their main naviga-

tion in HTML with PDF used for special pieces. Some sites, such as Adobe's own Web site (<http://www.adobe.com>), provide HTML and PDF versions of important pages. This is nice, but it is at least twice the work of PDF alone. Sadly, then, I conclude that for the time being, we shouldn't use PDF to the exclusion of HTML.

Summary

In this chapter you have learned something of the history of PDF and been introduced to each of the members of the Acrobat family of tools. You should by now be getting some ideas of the possibilities of PDF for Web publishing, and how it differs from traditional HTML Web publishing.

The next four chapters cover the tools and the possibilities of PDF in greater detail. Chapter 15, "Viewing PDF Files" will cover Acrobat Reader and viewing PDF files in more detail, while chapters 16 and 17 cover PDFWriter, Distiller, and Exchange in detail. Using the information in those two chapters and expanding on the ideas of optimizing for the Web, Chapter 18 turns to the topic of delivering your own stunning PDF files on the Web—and when not to!