

Table of Contents

[The Acrobat User](#)

Converting Web Sites to PDF Files With Web Capture

Acrobat's Web Capture feature allows you to convert a web page or an entire web site into a PDF file, with artwork, images, and links intact!

[PostScript Tech](#)

Programming Acrobat with *pdfmark* (Part 2)

How to add *pdfmark* calls that create bookmarks and links directly to documents in Quark, PageMaker, Microsoft Word, and other applications.

[Class Schedule](#)

February-March-April-May

Where and when are we teaching our Acrobat and PostScript classes? See here!x

[What's New?](#)

Acquired Knowledge closes its doors. Introducing Acumen Training and 5 South Software.

Sadly, AKI is gone. But you can still get the same excellent training and software development.

[Contacting Acumen](#)

Telephone numbers, email addresses, postal address, all ways of getting to Acumen.

[Journal feedback: suggestions for articles, questions, etc.](#)

Converting Web Sites to PDF With Web Capture

One of the cooler things you can do with Adobe Acrobat is convert a web page or an entire web site to a PDF file, complete with links and images. It's quite something to see: all of the html links are converted to PDF links to the appropriate page within the PDF file. If you click on a link that is still pointing to a web page (rather than a converted page within the PDF file), Acrobat can automatically append the new web page to your PDF file.

Quite cool.

But... Why?

I will confess to having been *very* skeptical about this feature when it first debuted. It seemed cool technology, but with no real reason for existing.

So I was wrong. I find I use this feature frequently. There is an awful lot of "how-to" documentation on the web. On-line documentation on everything from JavaScript to MySQL. Quite often I want this documentation on my laptop so I can refer to it when I'm on the road, teaching, without necessarily a dependable network connection.

With Web Capture, I can turn this html documentation into a PDF file. All of the links work; the fonts are embedded; the images are all in place. But it's now a single local file residing on my hard disk.

Very handy!

[Next Page ->](#)

For Example To see an example of a web page converted to a PDF file, click on the link below.

Warning: clicking on any of the links on that page will launch your web browser and take you to the corresponding html destination. That's on purpose; I didn't want to embed more than one PDF page in this newsletter. (I put a prominent "Go Back" button on the page that will bring you back here.)

[Show me](#)

This page is from the *printWriter.com* web site, one of my favorite prepress-related sites. To see the original web page corresponding to the Web-captured example, click [here](#). (Again, this will launch your web browser.) This page won't be quite the same as the web-captured page, simply because it'll no longer January 2001 by the time you read this; *printWriter.com* will have moved on to a new month.

[Next Page ->](#)

Capturing a Web Site

To convert a web site to PDF, go to the *Web Capture* submenu in the *Tools* menu and select the topmost item: Open Web Page...

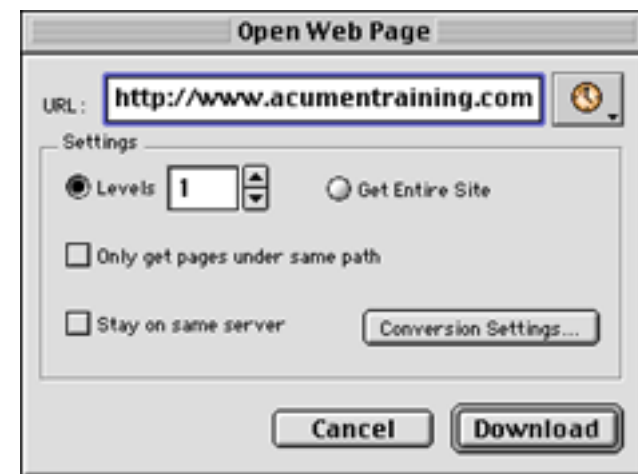
Acrobat will present you with the *Open Web Page* dialog box.

There are some reasonably straight-forward controls in this dialog box:

URL Type in the URL of the web page you want to capture.

Levels Indicate how to many levels deep you want Acrobat to follow links. A value of 1 tells Acrobat you want only this page. Any larger number will cause Acrobat to convert and add to the PDF file the targets of any links on this page.

Careful with this! Values greater than 2 or 3 can yield very large PDF files.



[Next Page ->](#)

Get Entire Site Convert *all* the web pages in the site to pages in the PDF file.

Warning: This is a very bad idea unless you know the site is very small. (Don't try this with, say, Microsoft's site.)

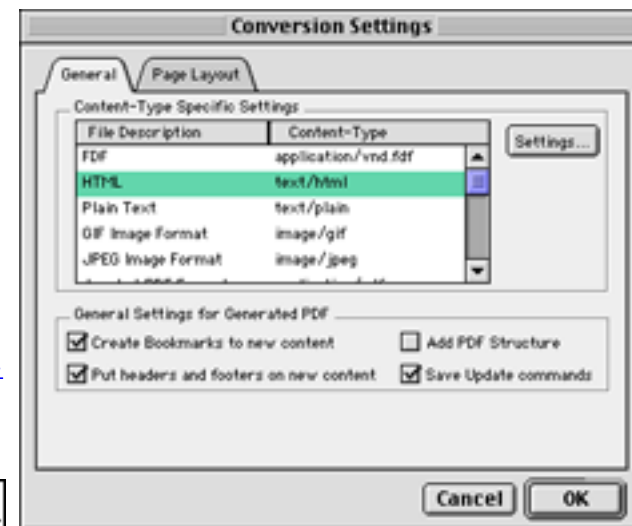
Only get pages under same path

Stay on same server

Both of these check boxes serve to limit the extent to which Acrobat will follow links to sites not directly associated with this web site. (If the site you're capturing has a link to Adobe's web site, you probably don't want to start slurping up all of Adobe's site and *its* links.)

Conversion Settings... This button leads you to a dialog box that allows you to specify details of the conversion. Explore its tabs and controls, and you'll find you can specify fonts, page size, margins, details of image conversions, etc.

[Next Page ->](#)



A Couple of Caveats

Although I've found Web Capture to be very useful, there are a couple of things of which you should be aware.

Budget Some Time

Web capture can take a very long time. If the web site you are trying to grab has even a moderately extensive network of links, you can find yourself trying to drink up many hundreds of pages of web pages, some of which are not directly related to your interests. Make use of those "Same path" and "Same server" buttons.

In any case, you should be prepared to start Web Capture and then go to lunch.

Don't Expect Miracles

Converting from html to PDF is not easy and, in many cases, it's impossible. Web Capture will ignore html features that have no PDF equivalent, such as frames and audio streams, and items that are not practical to capture, such as movies.

Don't expect to see this change anytime soon. This isn't something that needs fixing; there are simply limits to the extent that the one format can be converted to the other.

Still, It's Cool

Despite its limitations, Web Capture is worth using when appropriate. I like it.

[Return to menu](#)

Programming Acrobat With *pdfmark*, Part 2

When we last left our heroes, they had learned how to create bookmarks and links from within a PostScript file, using the Distiller-only operator, *pdfmark*.

This month, we'll finish up the topic by seeing how to insert these pdfmark calls directly into a QuarkXpress, PageMaker, Microsoft Word, or other document. The goal is to be able to distill these documents to a PDF file, complete with bookmarks and links already in place.

What we're going to do is place a pdfmark call into a hand-written EPS file. When you print the document to a PostScript file, your pdfmark snippet will be embedded in the PostScript output. Distilling the PostScript to PDF will create the bookmarks and links along with the content of the document, itself.

I strongly recommend you re-read first part of this article, in the December issue of the AKI Journal. (You can get that issue from the Acumen Training website by clicking [here](#).) I am going to presume in this article that you remember how *pdfmark* works.

[Next Page ->](#)

Handwritten EPS?

If you have taken one of our PostScript classes, you know that turning a PostScript file into an Encapsulated PostScript file is relatively easy. Assuming your PostScript file is well-behaved, you only need to add two lines to the very beginning of the PostScript:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 72 72
```

As a review, remember that these lines supply the following information:

%!PS-Adobe-3.0 EPSF-3.0

This line *must* be the first line of PostScript in an EPS file. This identifies the file as a PostScript file. The *3.0* values indicate to what version of the Document Structuring Convention and EPS specs the document conforms. Just use 3.0.

%%BoundingBox: 0 0 72 72

This reports where on the page the EPS file would print if you were to send it directly to a PostScript printer. As you may remember from your PostScript class, the numbers supplied are the *x* and *y* coordinates of the lower left and upper right corners of the graphic's bounding box.

The bounding box does *not* indicate where the graphic will print when you import the file into a document. At print time, the importing application will precede the EPS file's PostScript code with a *translate* and *scale* that changes the graphic to the correct position and size for the layout.

[Next Page ->](#)

Let's Make a Bookmark

Let's start with a simple case: an EPS file that creates a bookmark. We may import this EPS file into any document we wish in any application that supports EPS. When distilled to a PDF file, we will get a bookmark to the page on which we placed the EPS file.

Here's the code:

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 72 72

/pdfmark where
{ pop } { /pdfmark /cleartomark load def } ifelse

[ /Title (The Acrobat User)
  /OUT
  pdfmark
```

In the final PDF file, this creates a bookmark named "The Acrobat User." Note that this EPS file puts no marks on the final page; its only purpose is to create the bookmark.

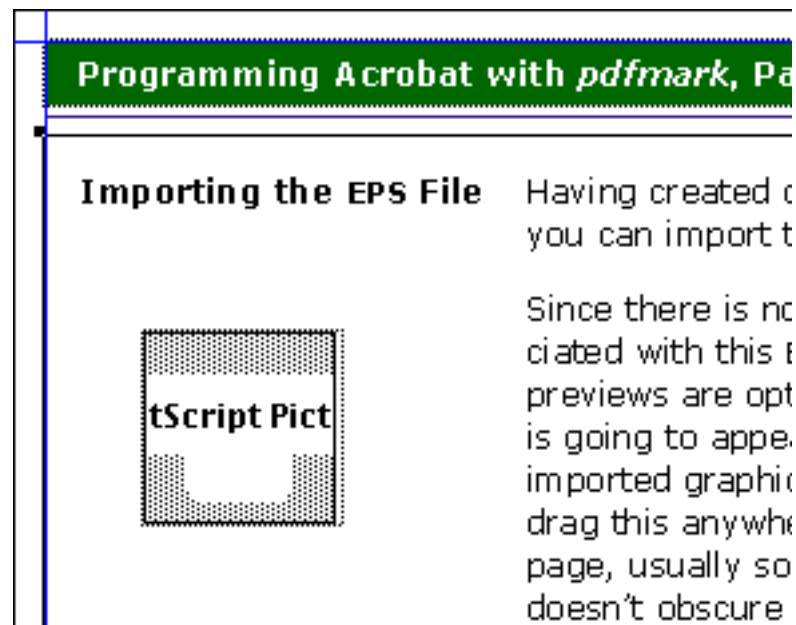
In the *%%BoundingBox* entry above, I'm supplying numbers for a one-inch square at the origin. This is pretty much arbitrary. Pick literally anything you want; it will have no bearing on the bookmark created by the EPS file.

[Next Page ->](#)

Importing the EPS File Having created our EPS file, we can import it into our QuarkXpress document. (Again, you can import this into *any* application that supports EPS graphics.)

Since there is no screen preview associated with this EPS file (remember, previews are optional in EPS files), this is going to appear as a generic imported graphic on our page. You can drag this anywhere convenient on the page, usually someplace where it doesn't obscure other page content. In the case of this newsletter, the Quark page ends up looking like the image at right.

I dragged the imported image into the left margin just to get it out of the way; its location has no effect on the bookmark it creates.



[Next Page ->](#)

Making Links

Creating links with this technique is just like making a bookmark: you create an EPS file containing a call to *pdfmark* that creates the appropriate link. What is different is that placement of the imported EPS file on the page now matters: this is the area that will be active when you distill to PDF; click on this area and the link will execute.

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 0 0 72 72

/pdfmark where
{ pop } { /pdfmark /cleartomark load def } ifelse

[ /Subtype /Link
  /Rect      [ 0 0 72 72 ]
  /Page      /Next
  /Border    [ 0 0 0 [ 0 1 ] ]
  /ANN
pdfmark
```

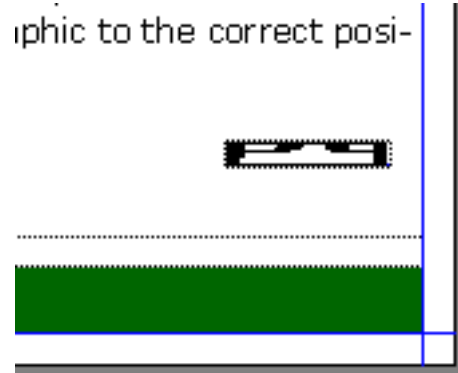
This call to *pdfmark* creates a link to the next page.

[Next Page ->](#)

Importing the EPS File When imported into a Quark document, the EPS file may be moved around and resized however I wish. The final position and size of the imported file will be the area that is “hot” in the resulting link.

All of the “next page” links in this journal are created with this technique. Each little, blue “[next page ->](#)” item on the page has an imported EPS file resting on top of it, as illustrated at right. This EPS file becomes a link to the next page when the file is distilled.

By the way, these EPS files *are* a bit annoying at times during layout; in my Quark document, the “next page” text on each page is obscured by the link-producing EPS file.



[Next Page ->](#)

A Couple of Notes In the sample code on page 5, the */Page* entry is associated with the name */Next*.

```
/Page /Next
```

This could also have been */Prev* or a page number within the PDF file. One minor nuisance is that Acrobat internally numbers pages from zero; this must be reflected in the number you supply here. The first page of the PDF file is *0*.

The *Border* entry as given produces no mark on the page.

```
/Border [ 0 0 0 [ 0 1 ] ]
```

This is surprisingly tricky. The first three values specify the *rgb* value for black. The nested array *[0 1]* is actually a dash pattern. If you took the PostScript Foundations class, you may remember that a dash pattern indicates the lengths of the alternating painted and unpainted segments in a dashed line. In this case, the length of the painted segment is *0*, indicating we should never paint the path.

[Next Page ->](#)

Conclusion

I've found this to be a terribly handy technique. The PDF files for my overheads and other documents all are created with links and bookmarks in place. It eliminates an great deal of fussing around with replacing pages and other techniques for preserving links from one version of a document to the next.

There is much I have not discussed regarding *pdfmark* in this article. Other *pdfmark* constructs that can usefully be placed into documents using a hand-written EPS file include articles, named destinations, and web links. I'll not be talking about those just now, however.

Perhaps later?

[Return to menu](#)

Schedule of Classes, Feb 2001 - Apr 2001

Following are the dates and locations of Acumen Training's PostScript and Acrobat classes. Clicking on a class name below will take you to the Acumen training website to the description of that class.

The PostScript classes are taught in Orange County, California, near the Orange County airport.

PostScript Classes

<u>PostScript Foundations</u>	Orange Co., CA	Feb 19 - 23	Orange Co., CA	Apr 2 - 6
<u>Advanced PostScript</u>	Orange Co., CA	Feb 26 - Mar 2	Orange Co., CA	Apr 23 - 26
<u>PostScript for Support Engineers</u>	Orange Co., CA	Mar 5 - 9	Orange Co., CA	Apr 30 - May 4

PostScript Course Fees PostScript classes cost \$1,750 per student

[Registration ->](#)

[Acrobat Classes ->](#)

Acrobat Class Schedule

Acumen training teaches three users' classes in Adobe Acrobat (the links below will take you to the Acumen website's complete description):

[Acrobat Essentials](#)

This class teaches the student how to make perfect PDF files. It includes complete coverage of the meaning and proper settings of all of the Distiller Job Options.

[Interactive Acrobat](#)

Here we show you how to add bookmarks, links, buttons, sounds, movies, form fields, and other interactive features to an Acrobat file.

[Troubleshooting with Enfocus' PitStop](#)

This class shows the student how to use all of the capabilities of this popular editing and preflight software.

On-site Only

The Acrobat classes are taught only on corporate sites. If you have an interest in any of these classes for your group, please see the Acumen website regarding setting up an on-site class.

[Back to PostScript Classes](#)

[Return to First Page](#)

Contacting John Deubert at Acumen Training

For more information For class descriptions or for any other information about Acumen's classes:

Web site: <http://www.acumentraining.com>

email: john@acumentraining.com

telephone: 949-248-1241

mail: 25142 Danalaurel, Dana Point, CA 92629

Registering for Classes To register for an Acquired Knowledge class, contact us any of the following ways:

Register On-line: <http://www.acumentraining.com/registration.html>

email: registration@acumentraining.com

telephone: 949-248-1241

mail: 25142 Danalaurel, Dana Point, CA 92629

[Return to First Page](#)

What's New at Acumen Training?

AKI Closes its Doors

After almost eleven years of conducting classes and developing software, **Acquired Knowledge** sadly went out of business on January 31. AKI would like to thank all of the people who have taken classes and supported its software over the years. Over the next month or two, AKI will be selling its assets, including its software. Those of you who are EZ-PDF or Download Mechanic users should watch for these products under new owners in the near future.

Acumen Acquires the AKI Curriculum

John Deubert, president of Acquired Knowledge (and author of this newsletter), has purchased the AKI curriculum and student materials. He will be teaching the PostScript classes regularly in Orange County, CA (near the Santa Ana/John Wayne airport). The Acrobat and PitStop classes will be taught on-site only.

5 South Software for Contract Development

Todd Donahue, vice-president of AKI, will be doing software development for Macintosh and Windows, the Web, and Adobe Acrobat as **5 South Software**. If you wish to contact him, please email him at todd@5southsw.com.

[Return to First Page](#)

Journal Feedback

If you have any comments regarding the *AKI Journal*, please let us know. In particular, we are looking for three types of information:

Comments on usefulness. Does the Journal provide you with worthwhile information? Was it well written and understandable? Did you like it, hate it, or did it make you want to drown your sorrows? How could we make it better? Do you like the PDF format?

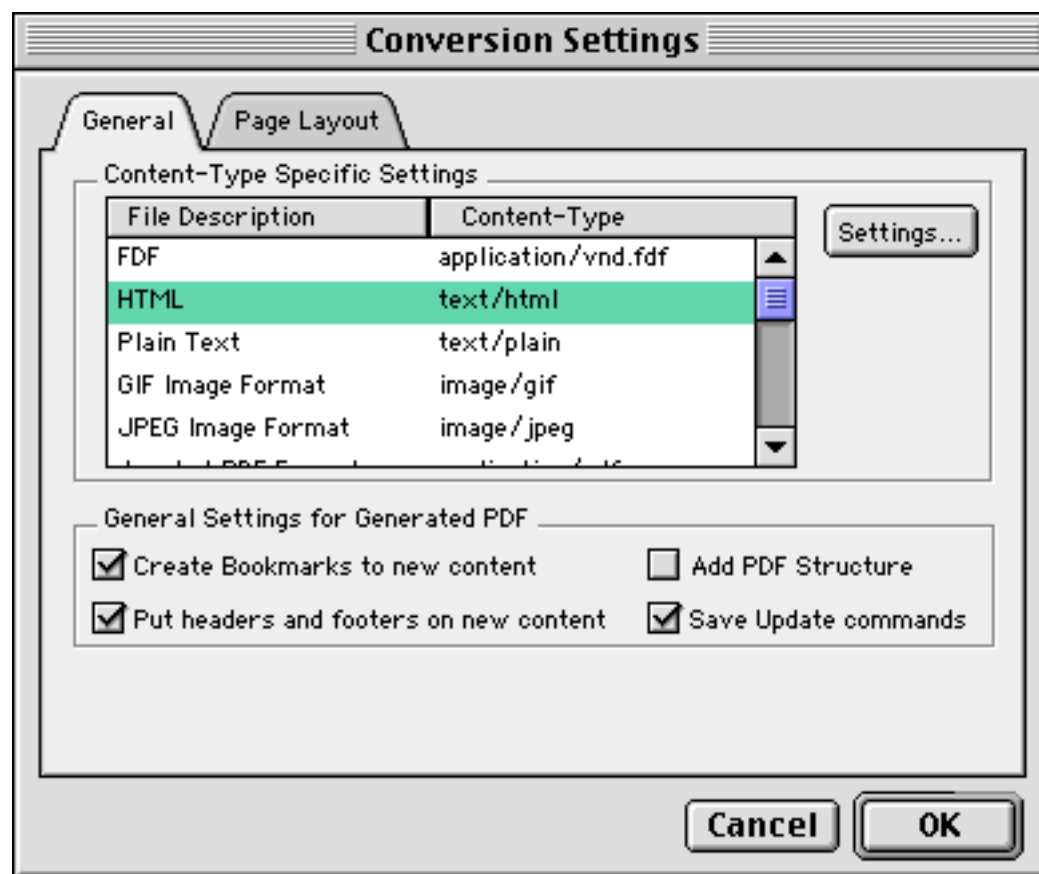
Suggestions for articles. Each Journal issue contains one article each on PostScript and Acrobat. What topics would you like us to address?

Questions and Answers. We are planning a Q&A section for future issues. Do you have any questions about Acrobat, PDF or PostScript?

Please send any comments, questions, or problems to

journal@acumentraining.com

[Return to Menu](#)





The Digital Technologies Network for the Community of Print



Search the Web:

Search

SEE MORE NEW ARTICLES WEEK OF JAN. 29

[Main](#) | [Commercial PrintWriter](#) | [PackagePrintWriter](#) | [Publisher Platform](#) | [Columnists](#) | [PODi](#)
[Intellectual Partners](#) | [Download Center](#) | [Research Center](#) | [Magazine Affiliates](#) | [HowIIPrepress](#)



[New Tech Tips Section !!](#)

[PrintWriter's HowToPrepress](#)

This update of PrintWriter.com debuts a new section, dedicated to technical information and tips on PREPRESS. The section, HowToPrepress, features technical tips, PDF talk and prepress profiles from the industry's technology community. In this update, see the following features and more: the popular PrintWriter column, **Tully Talk**; prepress information direct from the **International Prepress Association**, including talk from **IPA Forum**; X-Rite tips on calibration; tips on buying a scanner, scanning selection; a PDF profile of **Adobe Acrobat InProduction**. See an insightful column, care of IPA, on automating workflow, plus a comprehensive PDF detailing **Polaroid's Prediction** digital halftone proofer, and more.

[Xeikon's Digital Printing Outlook: Market Swimming to Success in 2001](#)

New print applications, from personalized catalogs to a variety of direct mail marketing projects to the full-color bathing suit catalog produced by a Xeikon press (at left), are elevating digital color printing. **Xeikon's Bob Barbera** views the growth of digital color printing as nothing short of explosive. With variable data applications growing in both acceptance and implementation, and the **future of web-to-print communication solutions** appearing by all accounts to be progressing at a healthy pace, it would certainly seem that Barbera's viewpoint is on the mark. Review this Q&A with the Xeikon executive, touching on issues for the commercial and package printing segments.

