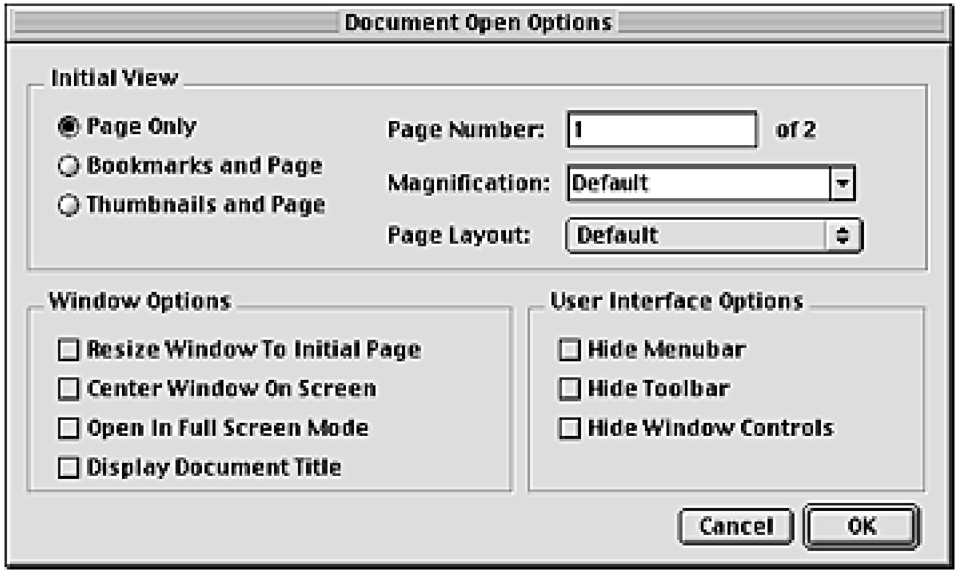
Table of Contents

Acumen
Training

"Document Open" Properties

Have you noticed how Acrobat behaves when you open an issue of the Acumen Journal? Acrobat resizes the document window to match the size of the PDF page, centers the window on the screen, and hides the toolbars.

The actions you want Acrobat to carry out when opening a PDF file are among the properties that may be stored as part of a PDF file. These "Document Open" properties are of great importance if your PDF will be read primarily on screen.


Document Open Options

Initial View

● Page Only
○ Bookmarks and Page
○ Thumbnails and Page

Page Number: 1 of 2
Magnification: Default
Page Layout: Default

Window Options

☐ Resize Window To Initial Page
☐ Center Window On Screen
☐ Open In Full Screen Mode
☐ Display Document Title

User Interface Options

☐ Hide Menubar
☐ Hide Toolbar
☐ Hide Window Controls

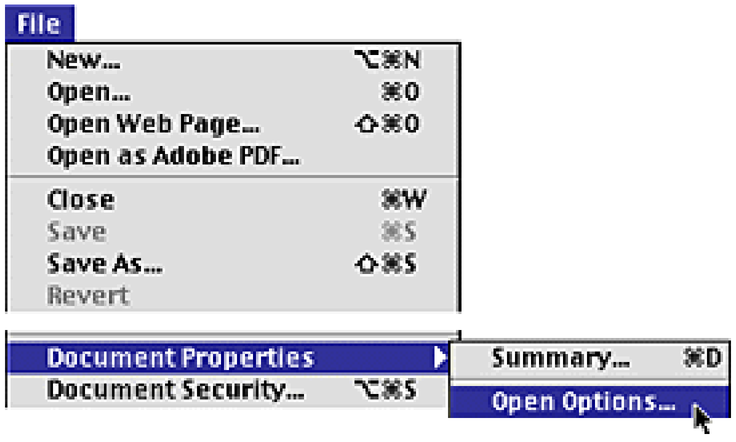
Cancel OK

This month, let's see how to use these Document Open properties.

Next Page ->

Selecting "Document Open"

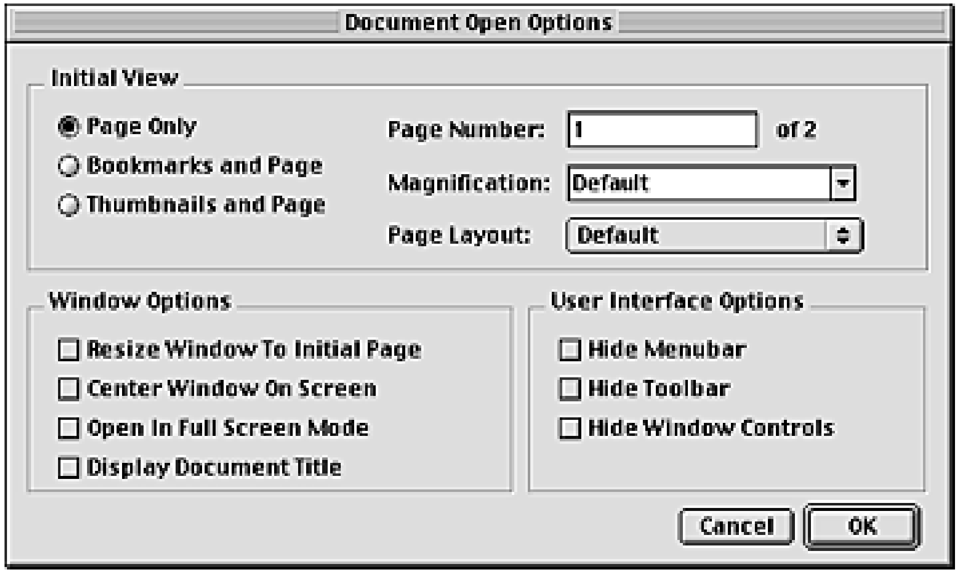
The Document Open properties reside among the "Document Properties" sub-menus. Simply go to *File>Document Properties>Open Options...* and Acrobat will present you with the *Open Properties* dialog box (below, right).



Here are listed all of the Document Open properties.

Let's look at each of these controls.

Next Page ->


Document Open Options

Initial View
- (●) Page Only
- ( ) Bookmarks and Page
- ( ) Thumbnails and Page

Page Number: 1 of 2
Magnification: Default
Page Layout: Default

Window Options
- ☐ Resize Window To Initial Page
- ☐ Center Window On Screen
- ☐ Open In Full Screen Mode
- ☐ Display Document Title

User Interface Options
- ☐ Hide Menubar
- ☐ Hide Toolbar
- ☐ Hide Window Controls

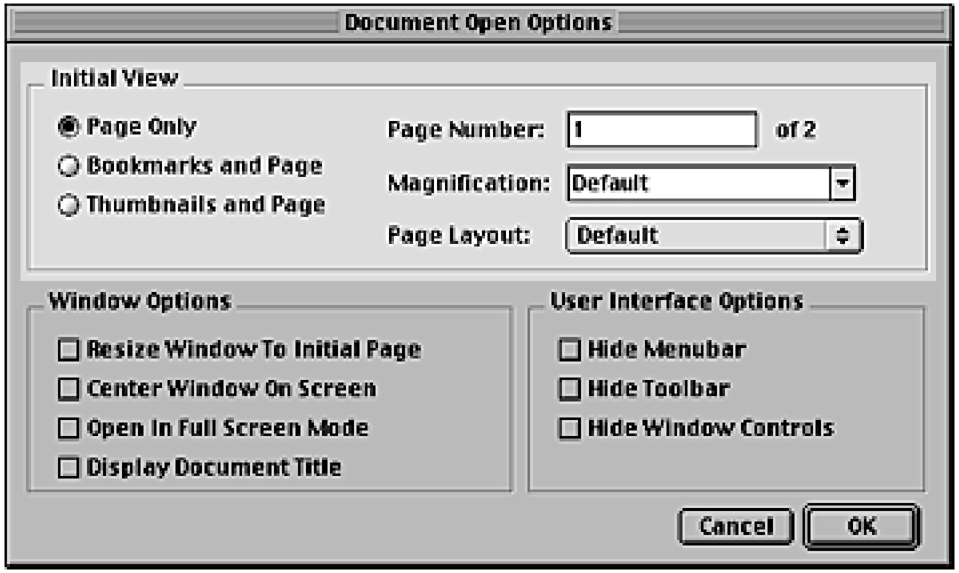
Cancel OK

Initial View

These controls allow us to specify how Acrobat should display the file when it is first opened.

Page Only
Bookmarks and Page
Thumbnails and Page

These radio buttons allow you to specify whether the PDF file's bookmarks and thumbnails should be initially visible.



Document Open Options

Initial View

[● Page Only] ○ Bookmarks and Page ○ Thumbnails and Page

Page Number: 1 of 2
Magnification: Default
Page Layout: Default

Window Options

☐ Resize Window To Initial Page
☐ Center Window On Screen
☐ Open In Full Screen Mode
☐ Display Document Title

User Interface Options

☐ Hide Menubar
☐ Hide Toolbar
☐ Hide Window Controls

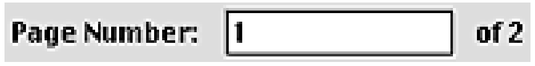
On-screen documents should always have highly visible navigation controls. The reader should never be left wondering what to do.

The first page of the Acumen Journal, for example, has a clickable table of contents that makes it clear to the reader where he or she may go. If your on-screen document doesn't do something similar, it is important that you make visible either the bookmarks or the thumbnails pane when the document opens.

By and large, bookmarks are much more useful than thumbnails for people who are reading your document for the first time. Thumbnails are really useful only for highly graphic documents being read by someone at least passingly familiar with the document.

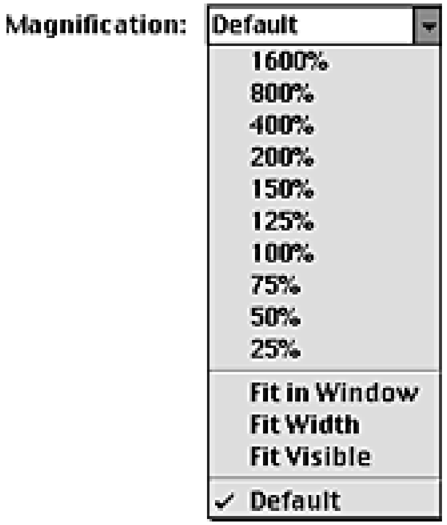
Next Page ->

Page Number



The *Page Number* text box allows you to specify the page to which the PDF document should open. This will be the first page of the document, as far as the reader is concerned. Usually, this will also be the first sequential page within the file, but it doesn't need to be.

Magnification



This control specifies the default magnification for this document. (That explains the name, anyhow.)

Most of the entries in this menu are simple numeric magnifications.

Fit in Window, Fit Width, and *Fit Visible* correspond to the equivalent controls in the toolbar. Be careful of using these as default magnifications for a document. The magnification your reader sees on the screen will be dependent upon how they have sized their windows; what they see may not be at all what you had in mind.

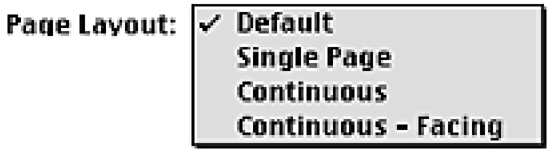
Default uses whatever the reader has set in Acrobat for the default magnification. Again, you will not be directly controlling the magnification, and so you can't be certain of what your reader will be seeing.

I recommend 100% for on-screen documents. If you need to set a different default magnification, it may be a sign that your document needs to be redesigned.

Next Page ->

Page Layout

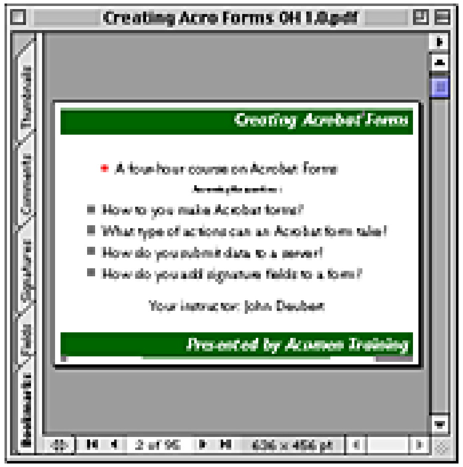
Finally, the *Page Layout* pop-up specifies how successive pages within your document should be displayed.



Single Page is the usual Acrobat method for displaying a PDF file: one page at a time. Clicking on the window's scroll bar moves you one page.

Continuous displays the pages as though they were on a continuous roll of film. Clicking the scroll bars move you up or down by a screenful, regardless of how many pages that might be.

Continuous – Facing is a sort of poor man's spreads: the pages are displayed continuously, side-by-side.


Creating Acro Forms OH 1.0.pdf

Creating Acrobat Forms

A four-hour course on Acrobat Forms

How to you make Acrobat forms?

What type of actions can an Acrobat form take!

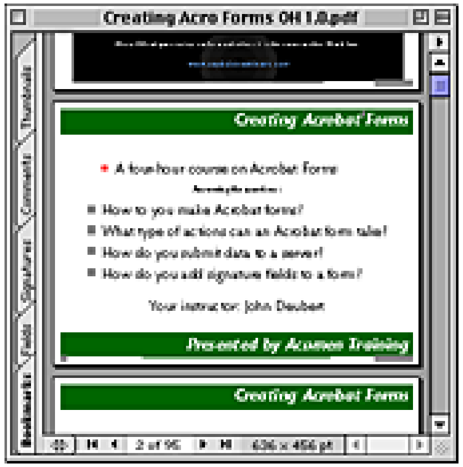
How do you submit data to a server!

How do you add signature fields to a form?

Your instructor: John Deubert

Presented by Acumen Training

2 of 95 636 x 456 pt


Creating Acro Forms OH 1.0.pdf

Creating Acrobat Forms

A four-hour course on Acrobat Forms

How to you make Acrobat forms?

What type of actions can an Acrobat form take!

How do you submit data to a server!

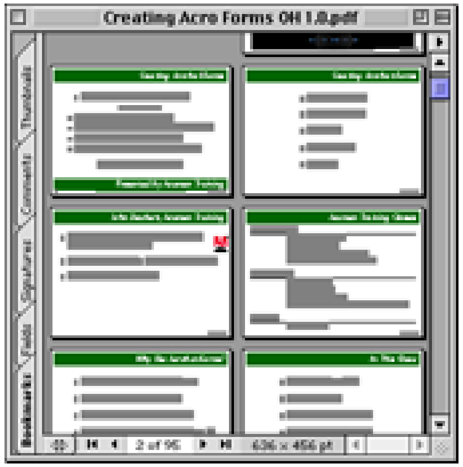
How do you add signature fields to a form?

Your instructor: John Deubert

Presented by Acumen Training

Creating Acrobat Forms

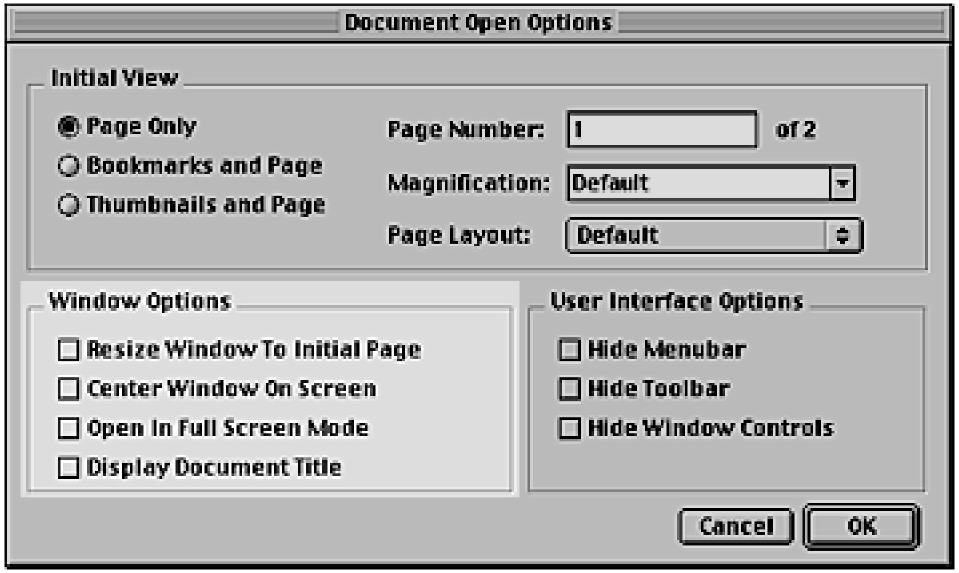
2 of 95 636 x 456 pt


Creating Acro Forms OH 1.0.pdf

2 of 95 636 x 456 pt

For on-screen documents, you will usually want to select *Single Page.*

Next Page ->


Document Open Options

Initial View

- ● Page Only
- ○ Bookmarks and Page
- ○ Thumbnails and Page

Page Number: 1 of 2

Magnification: Default

Page Layout: Default

Window Options

- ☐ Resize Window To Initial Page
- ☐ Center Window On Screen
- ☐ Open In Full Screen Mode
- ☐ Display Document Title

User Interface Options

- ☐ Hide Menubar
- ☐ Hide Toolbar
- ☐ Hide Window Controls

Window Options This series of check boxes specifies how the document window should be displayed to the reader.

Resize Window to Initial Page If selected, Acrobat will resize the document window to match the page size. An excellent choice for an on-screen document.

Center Window On Screen If selected, the PDF window will be placed in the center of the reader’s computer screen. Also a good idea for on-screen documents.

Open in Full Screen Mode If selected, the PDF document will fill the reader’s screen, hiding menus, controls, toolbars, etc. If you select this, be *very* sure to place controls on your PDF pages that let the reader regain use of the rest of his or her computer.

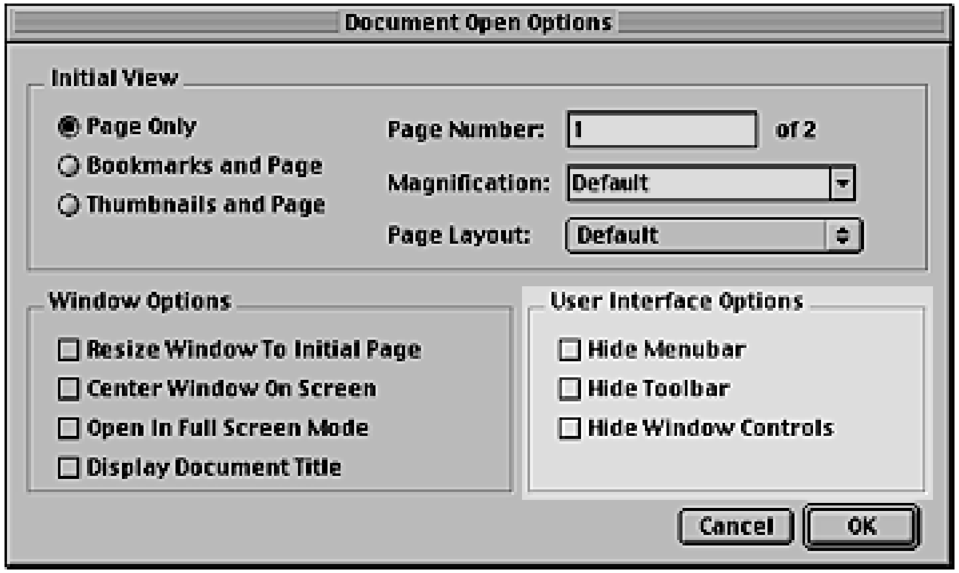
Display Document Title If selected, Acrobat will display the document’s title in the drag bar of the window. Usually a good idea, though this is more of an esthetic and document design question.

Next Page ->

User Interface Options

Finally, this set of check boxes lets you hide Acrobat's controls from the reader.

Be very careful with these. If you hide Acrobat's controls, you *must* provide good replacement controls within your PDF pages or you'll have a very panicky reader.


Document Open Options

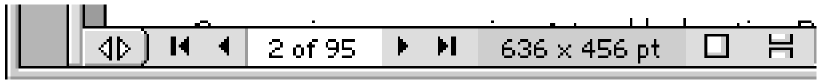
Initial View
- (● Page Only) ○ Bookmarks and Page ○ Thumbnails and Page
- Page Number: 1 of 2
- Magnification: Default
- Page Layout: Default

Window Options
- ☐ Resize Window To Initial Page
- ☐ Center Window On Screen
- ☐ Open In Full Screen Mode
- ☐ Display Document Title

User Interface Options
- ☐ Hide Menubar
- ☐ Hide Toolbar
- ☐ Hide Window Controls

Hide Menubar
Hide Toolbar
Hide Window Controls

These are the three things you can hide. Menubar and toolbar are pretty obvious. "Window Controls" refers to the page counter, scroll bar, etc. that normally run along the bottom of each Acrobat window.



Note that the Acumen Journal hides the Toolbar and Window Controls, but leaves the menubar visible. Even I tend to panic when the menubar disappears.

Next Page ->

Conclusion

Paying attention to the Open Option controls is an important part of creating an on-screen document in Acrobat. It is with these that you control the initial appearance of your document and the first impression that people have of you through your work.

Setting these appropriately go a long way to giving your PDF file a finished, professional first impact.

Return to Main Menu

Reencoding Fonts, Part 2

¿Dónde está el camino a San José?

Last month, we saw how to insert character names into fonts in order to gain access to the accented characters included in Adobe's Standard Character Encoding. The PostScript code we examined inserted four characters into a font, using a method that was clear, but not very efficient.

In particular, if you need to make extensive changes to a font's character encoding, you wouldn't want to do it a character at a time, as we did last month.

This month, we'll see how to do wholesale reencoding of a PostScript font.

By the way, you may want to go back and re-read last month's article. I'm assuming you remember how to reencode a font and why you would want to do so. If you don't have last month's *Journal* (and why not?), you can get it from the Acumen Training website; click here.

Next Page ->

Driver Output and Reencoding

When PostScript generated by an application or driver gets a font, it’s not enough to do a simple *findfont.* Driver output must reencode every font it uses. In particular, it needs to change the encoding of each font to match that of the operating system generating the PostScript (usually the Mac or Windows).

Thus, when driver output gets a font, the PostScript usually looks something like this:

```
/Helvetica findfont reencodefont 14 scalefont setfont
```

The *reencodefont* procedure will be a procedure defined in the PostScript prolog that, well, reencodes the font.

Next Page ->

***reencodefont* Definition**

Here is our *reencodefont* procedure. This program has the same output as last month's.

¿Dónde está el camino a San José?

```
/MacEncoding   [
     /.notdef /.notdef /.notdef
     ...
     ... Macintosh Encoding names here
     ...
] def

/reencodefont       % /NewName /OldName => <<fdict>>
{
     findfont dup
     length dict copy

     dup /Encoding MacEncoding put
     definefont
} bind def

/Helvetica-Mac /Helvetica reencodefont 18 scalefont setfont
72 600 moveto
(¿Dónde está el camino a San José?) show

showpage
```

Next Page ->

The Code in Detail

Define the Encoding array

```
/MacEncoding   [
      ...
      ... Macintosh Encoding names here
      ...
] def
```

We create our Encoding array ahead of time. This is simply a PostScript array with character names in an order appropriate to your operating system. To save you some typing, I've placed both the Mac and Windows Encoding arrays (as well as this month's sample program) in a file on the Acumen Training's Resources page. You're welcome.

Note that this Encoding array is appropriate only for Western Roman fonts. It will be completely wrong for Cyrillic or other non-Roman scripts and for fonts such as Symbol or Zapf Dingbats. We'll come back to this problem in a little while.

Define reencodefont

```
/reencodefont       % /NewName /OldName => <<fdict>>
{
```

Remember that to reencode a font, we create a new font identical to the original except for the new Encoding. Our *reencodefont* procedure therefore takes two arguments: the name of the original font and a name for the new, reencoded version.

Our *reencodefont* returns a copy of the reencoded font dictionary.

Next Page ->

```
findfont dup
```

Here we fetch the original font dictionary and duplicate the dictionary object on the operand stack.

```
length dict copy
```

We create a new dictionary as large as the original font dictionary and copy the contents of our original font into the new dictionary. This line leaves a copy of the new dictionary, now holding the contents of the original font, on the operand stack.

```
dup /Encoding MacEncoding put
```

We *dup* the new dictionary object and put our MacEncoding array into it with the name *Encoding.* This replaces the original Encoding array.

```
definefont
} def
```

Finally, we convert the new dictionary into a font dictionary. Note that *definefont* requires a name as its bottom argument. In our case, it will find the name that was placed on the stack as an argument to *reencodefont.*

Next Page ->

Using the new font

```
/Helvetica-Mac /Helvetica reencodefont 18 scalefont setfont
72 600 moveto
(¿Dónde está el camino a San José?) show
```

Finally, we use our new procedure to create a Macintosh-encoded version of Helvetica.

¿Dónde está el camino a San José?

Note that last month we printed the same text, but had to use backslashed octal character codes to print our accented characters:

```
    (\004D\001nde est\002 el camino a San Jos\003?) show
```

This month, I can type the characters directly into the string from my Mac's keyboard, because I am matching my font's Encoding array to my system's character encoding.

Next Page ->

Some Refinements

Non-alphabetic Fonts

As I mentioned earlier, our Mac- and WinEncoding arrays are appropriate only for Western, alphanumeric fonts; you would not want to use them with Symbol or dingbat fonts, nor with Cyrillic or other non-Western fonts.

Thus, before inserting an Encoding array into a font, you should check to see whether it's a proper font to receive this encoding. The way I usually do this is to check for the presence of *aacute* or some other character in the Adobe Standard Character set. I only insert the new Encoding array if my test character is present.

The proper way to test for the presence of a character in a font is to look in the font's *CharStrings* dictionary. Our *reencodefont* procedure now looks like this:

```
/reencodefont       % /NewName /OldName => <<fdict>>
{
     findfont dup
     length dict copy

     dup /CharStrings get /aacute known
     { dup /Encoding MacEncoding put } if
     definefont
} bind def
```

This is the version of *reencodefont* that is on the Acumen Training resources page.

Next Page ->

Caching Font Dictionaries

Reencoding a font can be time consuming if carried to excess; you don’t want to do it more often than necessary. One good way of minimizing the extent to which you execute your *reencodefont* routine is to do all your *reencodefonts* ahead of time, saving the result in a key-value pair.

```
/F1 /Helvetica-Mac /Helvetica reencodefont def
/F2 /Helvetica-Bold-Mac /Helvetica-Bold reencodefont def
```

While you’re about it, you should probably do your *scalefonts* ahead of time, also:

```
/F1-12 F1 12 scalefont def
/F1-14 F1 14 scalefont def
/F2-12 F2 12 scalefont def
/F2-14 F2 14 scalefont def
```

This way, when you need to change fonts, you don’t need to re-find, -encode, and -scale your fonts:

```
F1-12 setfont (You gave them ) show
F2-12 setfont (how much??) show
```

By the way… It would be still *more* efficient to place your scaled, reencoded font dictionaries in a composite font.

But that’s a tale for another month.

Return to Main Menu

Schedule of Classes, Dec 2001 - Mar 2002

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 description of that class on the Acumen training website.

The PostScript classes are taught in Orange County, California.

PostScript Classes

PostScript Foundations	December 10 – 14	March 18 – 22
Advanced PostScript	January 21 – 25	
PostScript for Support Engineers	January 14 – 18	
Jaws Development	April 2 – 5	

For more classes, go to www.acumentraining.com/schedule.html

PostScript Course Fees

PostScript classes cost $2,000 per student.
These classes may also be taught on your organization's site.

Registration →
Acrobat Classes →

Acrobat Class Schedule

On-Site Only	These classes are taught only on corporate sites. If you have an interest in any of these classes for your group, please see the Acumen Training website regarding arranging an on-site class.
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.
Creating Acrobat Forms	This class shows you how to make interactive forms in Adobe Acrobat. It steps you through creating the form, posting form contents to a server, and everything else you need to create a working PDF form.
Troubleshooting with Enfocus' PitStop	This class shows the student how to use all of the capabilities of this popular editing and preflight software.

Back to PostScript Classes

Return to First Page

Contacting John Deubert at Acumen Training

For more information	For class descriptions, on-site arrangements or 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 Acumen Training 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
Back issues	Back issues of the Acumen Journal are available at the Acumen Training website: www.acumenjournal.com/AcumenJournal.html

Return to First Page

Journal Feedback

If you have any comments regarding the *Acumen Journal,* please let me know. In particular, I am looking for two 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 eat brussels sprouts? 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?

Please send any comments, questions, or problems to:

journal@acumentraining.com

Return to Menu