Table of Contents

Journal feedback: suggestions for articles, questions, etc.

In-ʀɪᴘ Separations

PostScript can automatically create color separations from a composite color PostScript file. The separations can be for either process or spot plates, though the latter has some colorspace requirements.

Background

Printing a color art on paper entails overprinting two or more monochrome versions of the artwork, called *separations,* each printed with a different color ink, usually cyan, magenta, yellow, and black.

In traditional printing, separations are created by repeatedly photographing the original artwork through a series of filters, one for each separation.

In the electronic print era, documents are created electronically, resulting in files—images, ᴘᴅꜰs, etc.— that lend themselves to a variety of manipulations, including the production of color separations. Most (and perhaps all) professional image and graphics editing packages can convert a color image into a set of four black-and-white images, one each for cyan, magenta, yellow, and black ink. However, these packages generally have limitations; in particular, many of them cannot create separations from ʀɢʙ documents or images.

PostScript Level 3 introduced the capability for all PostScript printers to be able to print incoming print jobs as separations, rather than as composite color output. It's remarkably easy to do.

PostScript In-ʀɪᴘ Separations

When a PostScript program enables In-ʀɪᴘ separations in a PostScript Level 3 device, every page in the document will generate four pieces of paper (more or less; we'll talk about that in a moment), one for each ink needed by the document: cyan, magenta, yellow, black, and any spot inks. (We'll also talk about spot colors in a moment.)

The composite color PostScript document can use any kind of color in its page descriptions; in particular, PostScript has no trouble creating CMYK separations from RGB colors. (As always, some people may not like the way PostScript converts RGB colors to CMYK, but that's a separate problem.)

Turning on Separations

You turn on separations with a simple call to *setpagedevice* at the start of your print job:

```
<<   /Separations true
     /ProcessColorModel  /DeviceCMYK
>> setpagedevice
```

That's all there is to it.

There are two entries required in the page device dictionary:

*   *Separations* is a Boolean value that turns on separations (assuming its value is *true,* of course).

*   *ProcessColorModel,* tells PostScript what colorspace to use for process color separations. This is dictated by the press you are using; usually you are sending the separations to a four-color press, to be printed

with cyan, magenta, yellow, and black inks and so the appropriate colorspace is *DeviceCMYK*.

Additional Controls There are two additional key-value pairs you may place in the *setpagedevice* dictionary that affect separations:

SeparationColorNames These are the names of the plates you want, that is, the inks for which you want to create separations.

SeparationColorNames is an array of ink names:

```
<<   /Separations true
     /ProcessColorModel  /DeviceCMYK
     /SeparationColorNames [ /Cyan /Magenta /Yellow /Black /FrogGreen ]
>> setpagedevice
```

This says that you want to print plates for cyan, magenta, yellow, black, and FrogGreen inks.

"What's FrogGreen?" I hear you say. This is a spot color; we'll come back to it.

Note that you get the process inks for free, so if *SeparationColorSpace* is *DeviceCMYK* (it pretty much always is), then you can omit Cyan, Magenta, Yellow, and Black in the array.

SeparationOrder This is an array of ink names that indicates the order in which the plates should be printed. The order of the inks' names in the array is the order in which the separations will come out of the printer.

```
<<   /Separations true
     /ProcessColorModel  /DeviceCMYK
     /SeparationColorNames [ /Cyan /Magenta /Yellow /Black /FrogGreen ]
     /SeparationOrder [ /FrogGreen /Cyan /Magenta /Yellow /Black ]
>> setpagedevice
```

The above call to *setpagedevice* will cause the FrogGreen plate to come out of the printer first, followed by cyan, magenta, yellow, and black in that order.

SeparationOrder also specifies the particular plates that should be printed; only the inks whose names appear in the array will produce a page. Thus, the following *SeparationOrder* array:

```
/SeparationOrder [ /FrogGreen /Black ]
```

will print only the spot and the black plate; the cyan, magenta, and yellow plates will be suppressed.

Spot Plates

Let's talk about "spot" separations.

In print, a spot plate is a separation that will be printed with ink that is the actual color you want. Thus, the logo at right can be printed with two printing plates: one printed with black ink and the other with the appropriate purple ink.

A document that has only two or three colors in it is usually cheaper to print it as a series of spot plates, than as CMYK.

In PostScript, spot colors are defined with the *DeviceN* colorspace. Any DeviceN-based color can be printed on a PostScript Level 3 device as a spot plate by including the color's name in *SeparationOrder* array.

For Example Here's a PostScript snippet that would print separations for a black and a FrogGreen plate.

```
<<   /Separations true
     /ProcessColorModel  /DeviceCMYK
     /SeparationColorNames [ /Black /FrogGreen ]
     /SeparationOrder [ /FrogGreen /Black ]
>> setpagedevice

0 setgray                    % Print a black square
200 500 200 200 rectfill

[    /DeviceN                % Set DeviceN colorspace
     [ /FrogGreen ]          % Ink name: FrogGreen
     /DeviceRGB              % Alternative colorspace
     { 0 exch .75 mul 0 }    % Convert FrogGreen to DeviceRGB
] setcolorspace

1 setcolor                   % Select 100% FrogGreen
300 600 75 0 360 arc         % Draw a FrogGreen circle
fill

showpage
```
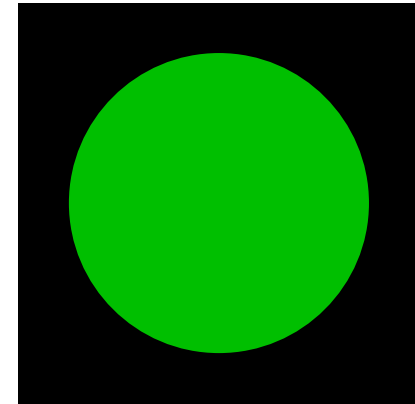
Note that the call to *setpagedevice* is the only indication that thi PostScript job should be printed as separations. The page description otherwise describes a composite color document without reference to how, exactly, it will be printed.

That said, it is important that the *FrogGreen* color be defined as *DeviceN* if we want to print it as a separate plate.

DeviceN, Very Briefly We don't have space for a complete discussion of the *DeviceN* colorspace; briefly, though, the colorspace array that we hand to *setcolorspace* has four components:

/DeviceN This specifies we want to use *DeviceN* for our colorspace.

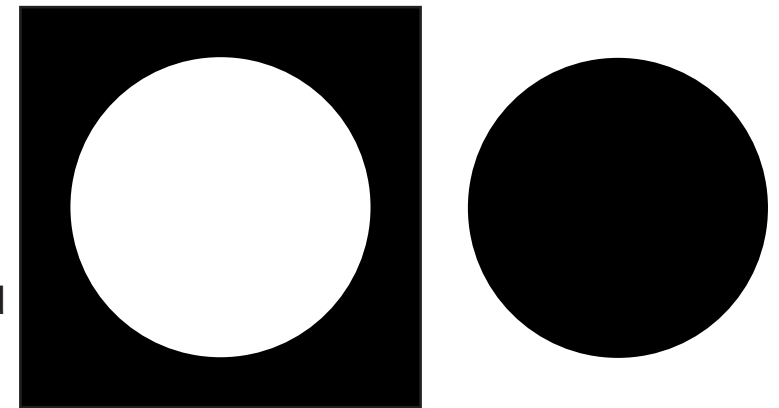[/FrogGreen] This is an array containing the names of all of the spot color inks we want to use.

/DeviceRGB This is the alternative colorspace. If the device on which you are printing or viewing the PostScript file (for example, your computer monitor) doesn't have FrogGreen ink or pixels, this colorspace should be used instead of *DeviceN*.

{ 0 exch .75 mul 0 }

This is the "tint transform," a procedure that converts a FrogGreen color value (on the range 0–1) into a color value appropriate to the alternative colorspace. This procedure converts FrogGreen into ʀɢʙ.

If we send this PostScript program to a Level 3 printer, it will print two black-and-white pages: one with a square-with-a-hole, to be printed in black ink, and another with a filled circle, to be printed in FrogGreen ink.

Note that any DeviceN colors whose names do *not* appear in the *SeparationColorNames* array will be printed as process separations and, therefore, appear to some extent on the cyan, magenta, yellow, and black plates.

Limitations

Nothings perfect, of course. In the case of PostScript's in-RIP separations, here are a few things to keep in mind:

- Not all PostScript RIPs support in-RIP separations. In particular, Acrobat Distiller does not. (I presume this was originally because Adobe didn't want to undermine the various PostScript color separation software packages.)

- Even if a particular RIP *does* support in-RIP separations, it may not be able to print very many separations at once. The RIP may need to allocate a page buffer for each plate it is rendering and it may simply not have enough RAM to accommodate all four (or five or six) plates in your separated print job.

Fixing Disappearing Hairlines in Acrobat 9

So here's the problem: you have designed a new, tasteful advertising panel for the local newspaper. Everything looks fine on your computer screen. However, when you print the document on your inkjet printer, all of the hairlines in the document disappear.

You check, double-check, triple-check the printer settings, but no matter what you do, the hairlines keep disappearing when you print.

So, what's happening?

The source of the problem is not obvious, but is easily fixed and Acrobat has a mildly obscure control that lets you fix it.
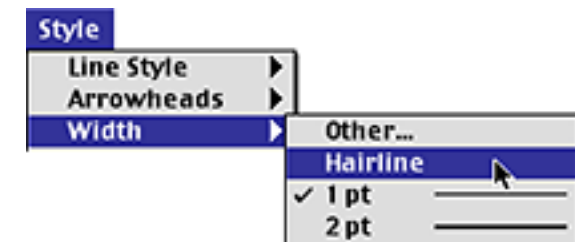
Let's see what's happening.

The Source of
the Problem

This problem mostly occurs in PDF files created with graphic, page layout, or other software that allows you to specify a linewidth as "Hairline," rather than a numerical value.

In this case, the software must decide what, exactly, constitutes a hairline, since the PDF file must specify linewidth as a numeric value. Surprisingly often, the software will specify a linewidth of *0* for its hairlines.

A zero linewidth has a special meaning in a PDF file: it specifies that the line should be the thinnest line possible on that printing device. In effect, this results in a line that is one pixel thick. This makes a pretty good hairline when viewed on screen and on a low-resolution printer: it's sharp, thin, altogether a satisfying hairline.

But on a high-resolution printer (say, 1200 dpi or higher), a one-pixel line is way too thin. Quite often, it is completely invisible.

Fixing It Happily, Acrobat Pro 9 can fix these too-thin hairlines. The software can scan through a PDF file, looking for lines whose widths are less than a threshold value; whenever it finds such a line, it sets the thickness to a minimum value. You have control over the threshold and minimum values.
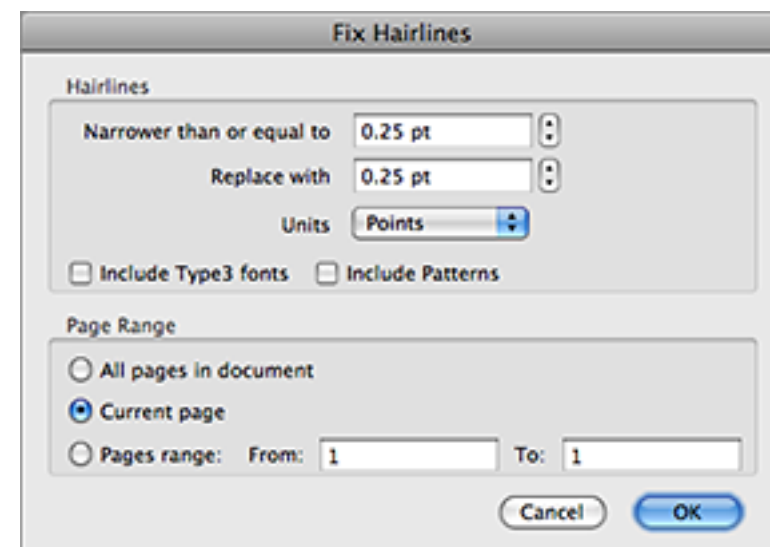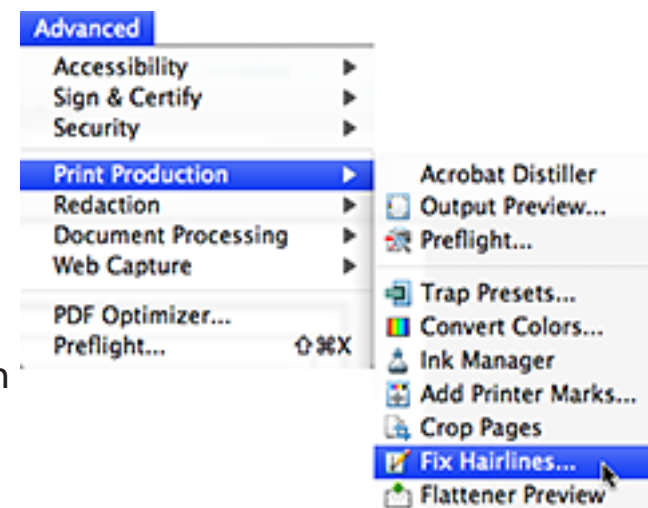
Here's how to do it:

1. Select *Advanced>Print Production>Fix Hairlines.*

   Acrobat will display the *Fix Hairlines* dialog box.

2. Select the parameters you want to apply (we'll discuss the controls in a moment).

3. Click the *OK* button.

Acrobat will scan through the PDF file, replacing too-thin lines with the minimum linewidth you specify.

Dialog Box Controls The controls in the *Fix Hairlines* dialog box dictate the behavior of the feature, as follows:

Narrower than or equal to...
This is the threshold linewidth. Lines thicker than this value will be ignored.

Replace with… This is the minimum line width for the document. Lines whose thicknesses are equal to or less than the threshold will be set to this width.

Units This pop-up menu lets you select the units you want to use in specifying the threshold and minimum thickness values.
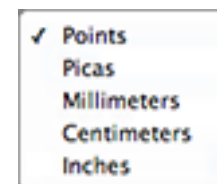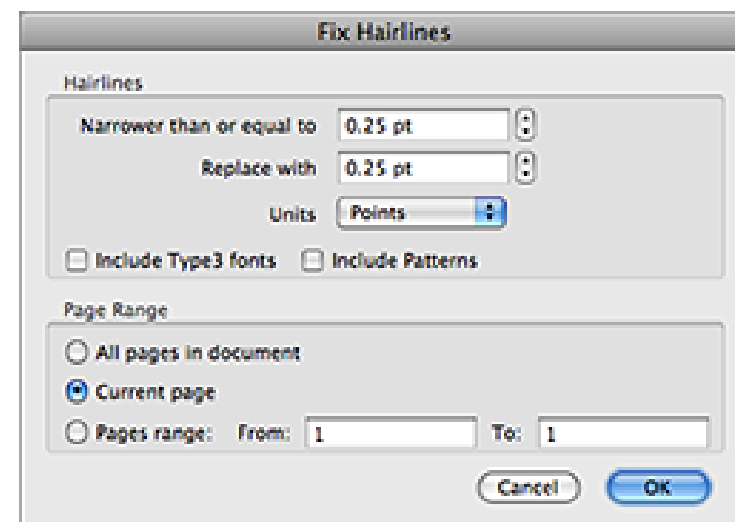
Include Type 3 Fonts
Include Patterns Type 3 fonts and PDF patterns define their character shapes and pattern tiles, respectively, using PDF drawing commands. They are therefore prone to the same thin-line problems as general graphics on the PDF page. These checkboxes tell Acrobat to examine characters and patterns drawn on the page and fix any excessively thin lines.

Page Range This group of controls lets you specify the set of pages that should have their hairlines adjusted. Your choices are :

- All the pages in the document

- The currently-visible page

- A specific range of pages

These choices are common and reasonably self-explanatory, so I'm not going to further describe their functions.

Limitations   The *Fix Hairlines* feature in Acrobat works very well, with one exception. There are some applications—notably some versions of Micorosoft Word—that draw their hairlines as long, exceedingly skinny, filled rectangles. These rectangles have the appearance of lines, but aren't.

I've no idea what madness led to the decision to make lines this way, but the technique makes the hairlines unfixable because there aren't actually any lines there at all, just a lot of aberrant rectangles. (Query: can anyone suggest a reason that Microsoft Word does it lines this way?)

Schedule of Classes, August – October 2009

At right are the dates of Acumen Training's upcoming classes. Clicking on a class name will take you to the description of that class on the [Acumen Training website](#).

These classes are taught in Orange County, California and [on-site](#) at corporate sites world-wide.

Please see the Acumen Training web site for more information, including an up-to-date schedule.

Class Fee Classes cost $2,000 per student, with the following exceptions:

- XPS class $1,500
- *Troubleshooting PostScript* $1,500
- *Support Engineers' PDF* $1,000

There is a 10% discount for signing up three or more students.

Note that if you have four or more students that need to take a class, it will almost certainly be cheaper to arrange an on-site class.

PDF Classes

PDF 1: File Content and Structure	Jul 27–30	Sep 28–Oct 1	
PDF 2: Advanced File Content			
Support Engineers' PDF	Aug 27–28		Oct 29–30

PostScript Classes

PostScript Foundations		Sep 7–11	Oct 19–23
Advanced PostScript			Oct 5–8
Variable Data PostScript		Sep 14–18	
Troubleshooting PostScript	Aug 24–26		Oct 26–28

XPS Classes (New!)

XPS File Content and Structure	Aug 4–6	Sep 22–24	

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: www.acumentraining.com   **email:** john@acumentraining.com

telephone: 949-248-1241

mail: 24996 Danamaple, Dana Point, CA 92629

Registering for Classes To register for an Acumen Training class, contact John any of the following ways:

Register On-line: www.acumentraining.com/register.html

email: john@acumentraining.com

telephone: 949-248-1241

mail: 24996 Danamaple, Dana Point, CA 92629

On-Site Classes Information regarding classes on corporate sites is available at www.acumentraining.com/Onsite.html. These courses are taught throughout the world; for additional information on classes outside the United States, go to www.acumentraining.com/OnsitesWorldWide.html.

Back issues All issues of the *Acumen Journal* are available at the Acumen Training website: www.acumenjournal.com/AcumenJournal.html

What's New at Acumen Training?

XPS Class Outline Now Available

A course description, including a day-by-day outline, of the *XPS File Contents and Structure* class is now available here on the Acumen Training website. The first class is now rescheduled for August 4 at Acumen Training's classroom site in Costa Mesa, California (near the Santa Ana/John Wayne airport). The three-day class will present in-depth coverage of the essentials of the XML-based Paper Specification.

See the website for the detail description, but the class includes such topics as:

- Open Packaging Convention
- XPS file structure
- Line Art
- Coordinate system
- Color
- Text
- Font Support
- Images
- Pattern fills
- Clipping
- Drawing curves

Like all Acumen Training course, *XPS File Structure and Contents* concentrates on those parts of the file format that apply to putting marks on a page. It is intended for

- Printer engineers working on devices that must consume XPS as a print stream

- Support engineers who support those devices

- Software engineers generating XPS for printed or displayed documents.

Don't hesitate to contact me to ask questions (or arrange an on-site).

I hope to see you there!

Journal Feedback

If you have any comments regarding the *Acumen Journal,* please let me know. In particular, I am looking for three types of information:

Comments on usefulness. Does the Journal provide you with worthwhile information? Was it well written and understandable? Do you like it, hate it? Did it make you think that perhaps global warming *isn't* the worst thing that can happen to you?

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

Questions and Answers. Do you have any questions about Acrobat, PDF, or PostScript? Feel free to email me about. I'll answer your question if I can. (If enough people ask the same question, I can turn it into a Journal article.)

Please send any comments, questions, or problems to:

john@acumentraining.com