PDF Color Secrets

By Bob Greene, Ph.D., Lantana

http://www.lantanarips.com

The PDF and PostScript information below has been published by Adobe and other sources. Because it is in technical manuals, almost no one reads it, or if they read it, they ignore it. It sometimes seems that Adobe employees also forget about these facts. This document will expose these de facto color secrets of PDF and their associated myths.

This article attempts to provide a clearer picture of what is really going on and related issues for composite color prepress workflows. Even though there are a few surprises, there is plenty of good news as well. Take heart! Don't get discouraged. "Do's and don'ts" are scattered throughout this article to help you get the results that you require. Also a few questions are left for the reader to ponder.

Color spaces

Acrobat and PDF support objects specified in a number of device dependent, device independent, and special color spaces. Except for certain controlled environments, production professionals should assume that different color spaces may be intermingled in any document, perhaps in any page, particularly if you have graphics or pages included from different sources. The trick is to understand what is going on and to plan accordingly.

Color spaces in PDF and Acrobat refer to that used when the object was created within a PDF. Device color spaces refer to the color space used by the creation device. For images, this would often be the color space when the data leaves the scanner. But it could also be the color space applied in Photoshop or Quark, or the result of color conversion by the printer driver, whichever made the last modification. Normally, all these programs go to great lengths to preserve colors. With care, device dependent color reproduces quite well. But that is also its weakness. That care is manifested as art, craftsmanship, or cross-organizational communication - items that are becoming more scarce in our fast-paced, de-centralized world.

Fear and reality have led to the desire for color management throughout these processes. The reality is that the scanner data might not display the same colors on a monitor or printer. This has led to the notion that you can attach a profile that will provide the information to convert from the source colors to the colors specific to the destination device. But if you attach a profile that makes the colors look right on the monitor, then it probably won't look right on a printed page, and vice versa. To get around this dilemma, some color management systems will have one profile for each destination device or printing condition (print technology, paper, ink) per each source device. If you had M source devices and N destination devices or print conditions, this could lead to M x N profiles to construct and manage.

Increasingly, you don't know where your files will be printed or on what type of device. Often, the print buyer decides this at the last minute, based on the best price, delivery schedule, and destination. There is no guarantee that you are applying the right profile and that you will get the intended colors.

To get around all this, Adobe, Apple and a few other companies proposed an industry standard for device independent color. Initial implementations included the CIE-based device independent color in PostScript Level 2 and Apple's ColorSync. The group and the effort eventually became the International Color Consortium (ICC).

The premise of ICC color is that you can use a profile to map source colors into a mathematically-rigorous absolute color space and have other profiles to map from this absolute color space to destination colors. This de-couples the creation and rendering of color information. You only need apply the appropriate profile at the destination. Also, it means that M + N profiles are needed, instead of M x N profiles needed for traditional color management, that is, many fewer profiles.

ICC color also recognizes that the same colors can be applied different ways. This has led to the notion of "rendering intents" for different objects. Four standard rendering intents are defined. Others are possible. The table below lists these and some of their properties.

ICC, PDF & PostScript Rendering Intents

Rendering intent	Objective	Best for	Advantages	In-gamut mapping	Out-of-gamut mapping
Absolute colormetric	Color matching	Logos, color catalogues	Measures the same across media	Exact	All to in-gamut nearest color
Relative colormetric	Media adjusted color matching	Vector graphics	Looks the same across media	Exact, but white point adjusted	To nearest in-gamut color, white point adjusted
Saturation	Preserve saturation	Business graphics	Vivid	Exact sometimes	To nearest, most saturated in-gamut color
Perceptual	Preserve color relationships	Scanned images	Pleasing	Compressed relatively within in-gamut region	Compressed relatively into in-gamut region

© 2000 Lantana Research Software Corporation

Data manipulation

Color conversion is not always exact. There is the chance that you will have some round-off error, particularly for the more sophisticated mathematics involved in ICC color. So, you can't always expect to end up with exactly the same color after doing a conversion and its inverse. Therefore, the fewer the number of conversions the better.

Separation logic, i.e., conversion into CMYK also can lead to surprises. Even under perfect conditions, you might find cases where what you believe is a single color separates onto two or more plates.

Particularly for high key images, you might find dots on the black plate when you did not expect them. This can be exacerbated by the choice of image compression. This is generally not a problem for lossless compression where you get back exactly what you started with after a compression-decompression cycle. It is more likely to happen with lossy compression, such as JPEG.

Putting it into practice

The accompanying table shows a comparison of all the color spaces supported in PDF and PostScript and how they relate. For every PDF color space, there is an available PostScript color space. This also explains why PostScript is the output file format for graphic arts applications. A corresponding chart for PCL would have too many gaps.

PDF & PostScript Color Spaces

PDF Color Space	Type	1st PDF Vers.	1st Acro Vers.	Equivalent PostScript Color Space	1st PS Level	Comments
DeviceGray	D	1.0	1	DeviceGray	1	
DeviceRGB	D	1.0	1	DeviceRGB	1	
DeviceCMYK	D	1.0	1	DeviceCMYK	1.x	Late addition to PS 1, formally added in PS 2. Level 1 & 2 separation mechanisms are different.
CalGray	I	1.1	2	CIEBasedA	2	
CalRGB	I	1.1	2	CIEBasedABC	2	CIEBasedABC can cover any 3-component device independent color space, e.g., CalRGB, Lab, etc.
Lab	I	1.1	2	CIEBasedABC	2	
Separation	S	1.2	3	Separation	2	For spot colors. Crop marks etc. are done in Separations=All to appear on all plates.
Indexed	S	1.2	3	Indexed	2	Table for a fixed number of input values. All but Pattern and Indexed may be indexed.
Pattern	S	1.2	3	Pattern	2	
DeviceN	S	1.3	4	DeviceN	3	For duotones, HiFi color seps, etc. Appears as composite color in Acrobat.
ICCBased	I	1.3	4	See Comments		A PostScript 2 or 3 CIE color space with the same number of components may be used.
Unavailable	I			CIEBasedDEF	3	Permits 3-component table look-up before normal CIEBasedABC processing.
Unavailable	I			CIEBasedDEFG	3	Permits 4-component table look-up before normal CIEBasedABC processing. Good for CMYK to CMYK conversion.
Color Space Types: D = device dependent, I = device independent, S = special						

© 2000 Lantana Research Software Corporation

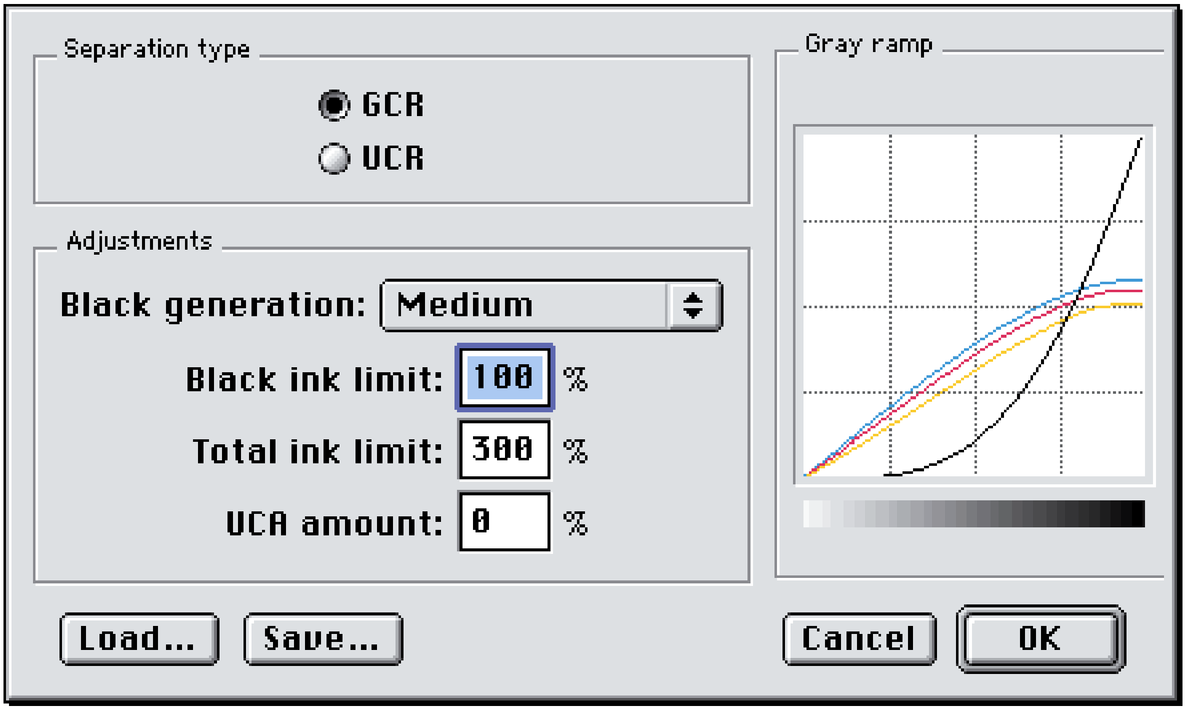
What is the best color space for production work? No matter which you choose, you should consider what steps you will be taking downstream from Acrobat. If you are printing directly from Acrobat, life is simpler. But if you are outputting PostScript files to other downstream processes, such as trapping or imposition, then you need to verify that each of these adhere to the same assumptions.

As stated before, different color spaces may be used within a page. But for someobjects, you have no color space choice. True PostScript 3 duotones, tritones and quadtones are specified in DeviceN. Spot colors are represented using Separation color space. Optionally, either of these color spaces may be printed to separate plates or converted to process color on output. But what about RGB versus CMYK versus ICCBased?

RGB versus CMYK

If you are preparing documents solely for the web, then for the same number of image samples, RGB is negligibly faster with possibly smaller file size. If you are preparing documents for print, then CMYK is better, since it preserves spot colors, and separating RGB may be unsatisfying or not possible. But if you are preparing documents for both web and print, CMYK is the better choice. CMYK documents coexist quite nicely in the RGB world, but the converse does not always hold true. And there are ways to convert high-resolution PDF images into low-res for the web.

But you can't always choose CMYK. Many documents originally created using Microsoft or other PC applications, arrive in your shop as RGB. Your choices are basically two: 1.) immediately convert them to CMYK, using Lantana's PDF ImageWorks, Enfocus PitStop, or Quite's A Box of Tricks, or 2.) rely on downstream processes, such as, in-RIP separations to convert RGB to CMYK. To improve this latter case and to ensure that RGB black text converts to the black plate only, Lantana's Crackerjack plug-in to Acrobat offers UCR/black generation (GCR) as part of its suite of functions to control PDF output to high-end printing devices.



Separation type

● GCR

○ UCR

Adjustments

Black generation: Medium

Black ink limit: 100 %

Total ink limit: 300 %

UCA amount: 0 %

Crackerjack UCR/GCR Panel

Monitors, scanners, digital cameras, etc. deal in visible light. The color spaces to express these (RGB or various ICC color spaces) are designed to include all visible light. But printing deals in light reflected off or through inks, which allow a narrower color range, or color gamut. An on-going issue is how are more visible colors converted into the fewer colors of the CMYK print gamut. See theRendering Intent table above for more on this subject.

About ICC

There is a mathematical beauty to ICC color. But the way that ICC is converted to CMYK is defined through the profiles and the tools that create them. With “canned” profiles, there is no guarantee that gray text will be constrained to the black channel. If you don’t believe this, you can try it yourself in Adobe Photoshop. In Lab, select a gray value (a=b=0). Then look at the CMYK equivalent. Odds are that you have non-zero values for all CMYK channels.

Do you need ICC color for everything? Perhaps not. Computer generated text and art is created with precise specification of color, i.e., in a device independent way, usually something like “give me 50% green and 30% red”. But it supplies a precise definition. Most device dependencies are a result of scanning. So, if you used the same profile for text, graphics and images, you are probably unintentionally altering the color of the text and graphics.

Processing ICC data can have other interesting effects. One PDF and PostScript expert points out that overprints may be lost in profile to profile conversion, but that traps should be preserved. Theoretically, there should not be a problem here. Perhaps he is speaking from practical experience. The bottom line is that this is another thing for which to watch.

Don’t be daunted. ICC-based color management is a discipline, as is all color management. You need to establish proper procedures. But you can get good results. You will want to conduct test runs with steps wedges of primary and secondary colors. Based on the results you will want to tweak your profiles with the appropriate tools. Done properly, color-managed PDF should give you increased color reliability and reproducibility across time, location, and output device.

Creating a PDF

For any document that may be professionally printed, from your creation application you should embed all fonts and 1.) create a PostScript file and process it through Acrobat Distiller, or “Export to PDF” properly configured , which performs the same function. You should always use the Distiller PPD for this process, but make sure its DefaultColorSpace is set to CMYK. If it is set to RGB, then some objects will be created in RGB in the PDF. Also, if you use an imagesetter PPD set to Gray, then some color image formats will be created in grayscale in the PDF.

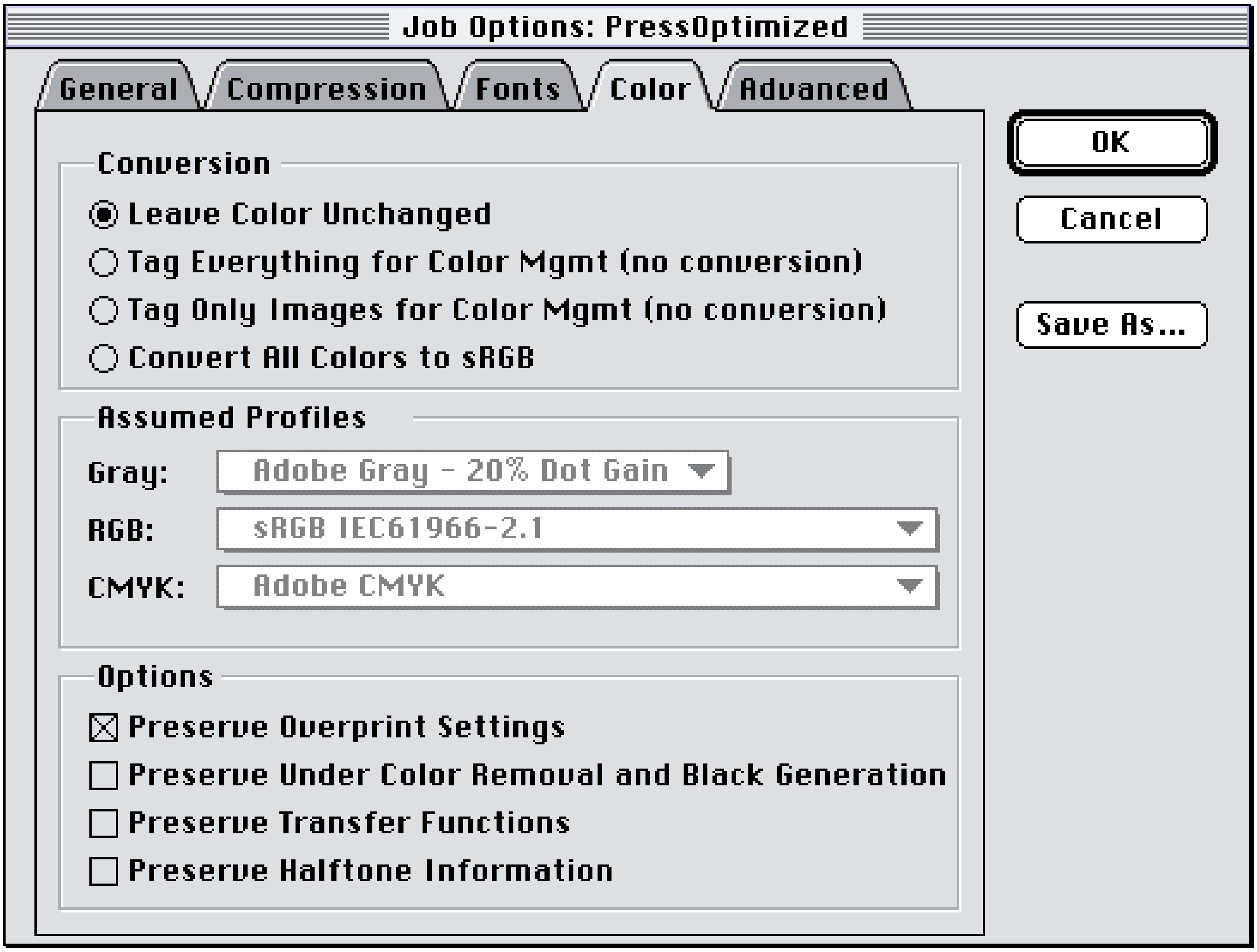
Does the PostScript level make any difference when preparing a file for distilling? Yes, it does. As you can see from the PDF & PostScript Color spaces table above, certain applications, such as duotones or spot color gradients can only be satisfied by PostScript 3 and PDF 1.3. In general, for maximum application capabilities, you want to create PDFs in the latest version. Naturally, you need to have the latest version of Acrobat to go with it. Crackerjack will allow you to print these on a Level 2 device as well, to the best of the RIP's capabilities.

Much has been written elsewhere about various Distiller 4 color parameters. But it does not hurt to review a few points. You should start with the "PressOptimized" settings. With a few exceptions, Adobe has made pretty good decisions overall. You can make new settings to meet your needs.

First, a look at compression. Adobe has specified "Automatic" compression at maximum quality, which usually equates to minimum compression and larger file size. The automatic compression chooses ZIP for images with sharp transitions and JPEG otherwise. JPEG is a "lossy" compression, that is, a round trip gives you different data from that with which you started, but they look equivalent. If the data is 3-component, such as RGB, it also is transformed to separate luminance from chrominance. Adobe is trying to do the right thing, i.e., preserve the black channel. But there are no guarantees that transformation, compression, and decompression will be 100% successful. To avoid this transformation, change the selection to "ZIP". If greater compression is more important, select "JPEG" and a different quality level.

PressOptimized specifies "Leave Color Unchanged" for Distiller's PDF color creation. This is the clear choice if you are trying for a traditional CMYK workflow. The other options are variants of a color-managed workflow, and allow you to select ICC color profiles. Of these, I prefer "Tag Only Images for Color Mgmt (no conversion)". With this text and graphics, which usually have precise color definitions anyway, will not be altered. Also, this selection allows black text to remain defined as CMYK black.

The default profile selection, "Adobe Gray - 20% Dot Gain", is Adobe's second worse idea on this panel. This choice adjusts only one type of data for a case that may or may not apply. The good news is that it is limited to grayscale images. The problem with this will become more obvious as we discuss transfer functions below.



Conversion

● Leave Color Unchanged
○ Tag Everything for Color Mgmt (no conversion)
○ Tag Only Images for Color Mgmt (no conversion)
○ Convert All Colors to sRGB

Assumed Profiles

Gray: Adobe Gray - 20% Dot Gain
RGB: sRGB IEC61966-2.1
CMYK: Adobe CMYK

Options

☒ Preserve Overprint Settings
□ Preserve Under Color Removal and Black Generation
□ Preserve Transfer Functions
□ Preserve Halftone Information

Adobe Acrobat Distiller Color Job Option Panel

The other color options address overprint, UCR, transfer functions, and halftone functions. There is no disagreement about "Preserve Overprint Settings". This has positive benefits for text variations in multiple languages and composite color traps created upstream. "Preserve Under Color Removal and Black Generation" provides instructions for a PostScript RIP to convert RGB to CMYK with varying degrees of rich black. This is really a user preference. Crackerjack provides the same capability, but applied downstream for all the pages in the job.

Adobe's worst choice on this panel is selecting as the default, "Preserve Transfer Functions". There used to be an "Apply" option in Acrobat 3. This made much better sense. Perhaps it will reappear in Acrobat 5. Transfer functions, until changed, affect all succeeding data. It is common for pages to be composed in difference locations, merged into a single PDF document, and imposed on to the same flat. This choice has the potential to pervert the appearance of pages from different sources. It should normally be "Off". It should be "On" only when trying to achieve artistic effects for single page output.

Why are transfer functions there anyway? The best I can determine is that they are often there to compensate for the fact that Photoshop CMYK JPEG files are the inverse of normal CMYK JPEG files. The simple workaround is to use TIFF format instead. Other cases seem to center around PageMaker trying to hide undesired elements. Those can simply be deleted.

There is no disagreement about “Preserve Halftone Information”. Generally, this should be “Off” since it can be specified consistently for the entire job when going to the RIP, as in Crackerjack. Having all the same halftoning will minimize moiré on the press. You might turn this option on if you want to preserve halftone information when reproducing special effects for single page output.

Distiller’s “Advanced” panel has one other parameter related to color, that is, “Preserve OPI Comments”. Adobe’s choice of being “On” has the side effect that TIFF images are created in PDF as Form XObjects rather than the simpler Image XObjects. Some Acrobat plug-ins have difficulties interpreting these more complicated objects. So if you don’t use an OPI workflow, turn “Off” this selection. Conversely, if you use an OPI workflow, remember that only the low resolution preview is placed in the PDF. Any color operations performed in Acrobat or by plug-ins are not performed on the high-resolution file that is actually printed.

There is one other ambiguous aspect to OPI. With color management, you are “tagging” the images. But if you are also using OPI, is your tag for the preview only? Or will it also apply to the substitute high-resolution image? In PostScript, often the color space definition is within the “image object” but before the image stream. If your OPI server replaces just the image stream, then the profile will apply to the high-resolution data as well. But if the OPI server replaces the entire “image object”, then the profile may be lost on substitution.

Colorized bit maps or “fake duotones” is another special case. Users will place a grayscale TIFF image into QuarkXPress, apply a color to the placed image, and set the background of the picture box to a different color. On screen, the color background shows through the colored image. But for composite printing workflows, the image knocks out the background and spot colors are often converted to process. Creo/Scitex has developed a free Quark extension, Prinergy Print XT, to “legalize” this ad hoc approach. The extension is available at http://www.prinergy.com. After Distilling, the PDF contains DeviceN color space commands.

Occasionally, one wants to create a two-color document. Particularly with Microsoft applications on the PC, one selects from their color palette. Many people incorrectly assume that those are spot colors and are specifying a spot color and black. Not so! The palette colors are RGB. So they end up with data on black plus at least two other plates. If available, use PitStop Professional (see below) to globally convert the second color to spot. Another approach is to use colors in the creation application that are close to ink primaries, or as Microsoft calls them, pink or turquoise for magenta or cyan, respectively. When you output the resultant PDF with Crackerjack, you can get two fully saturated separations. You can use any ink you like for the second color when you run it on a press.

Life as a PDF

This whole article centers on a composite color workflow. But what if you are forced to deal in pre-separated files, i.e., where each page is a separated plate and appears gray in Acrobat? A new Acrobat plug-in, named PDF Seps2Comp from Intense Software, allows you to recombine separations, even trapped files and negatives, into composite color PDF pages. This approach can also address PDFs from DCS and copydot files that were first placed in Quark. What if the PDF document were tagged for color management when created in Distiller? This is probably a pretty unusual case. But since each separation comes in as a gray page, the “gray profile” would normally be applied. You might want to change it to a “CMYK profile”.

Altering color in a PDF falls to a few trusty tools. PitStop Professional is the best tool for preflight, correction and editing PDF text and graphics. It currently supports nearly all of the PDF color spaces, and comes in handy for global color changes.

Images are best handled by PDF ImageWorks which provides numerous tools for editing image data and managing whole images, such as, image substitution or extraction. It is also the best tool for resampling PDF images, as it provides subsampling and a number of downsampling algorithms. Image size is the single biggest difference between a document for print and one for the web or CD.   For these applications, it is not necessary to convert CMYK to RGB to be web-ready. CMYK works quite nicely in Acrobat Reader.

There are three cases for trapping page data; before it is a PDF, when it is a PDF, or after it is a PDF. Before it is a PDF comes in two basic flavors, composite color traps, from applications such as PageMaker, or pre-separated trapping from Quark that has been recombined by PDF Seps2Comp. The objective is for all data to comply to the same standards, in this case, composite color. The second case is trapping a PDF. This might be accomplished by applications, such as Illustrator, that can open a PDF, trap, and then save the trapped file as PDF. Ideally, one wants to trap directly in PDF. Hopefully, we will have that some day. Part way to the third case are Acrobat plug-ins, such as the one downloadable from Agfa and also distributed with Apogee, that allow you to set the trap parameters and page zones for in-RIP PostScript trapping. The other variations of the third case are downstream PostScript trapping applications. These applications should be PostScript 3 fluent to support all the various PDF and PostScript color spaces. Examples of these packages are Ultimate Technographic’s Trapeze and ScenicSoft’s TrapWise.

Another curious state exists when you try to edit a PDF image or graphic in Photoshop or Illustrator. What happens when you edit a color managed document on a workstation where color management is turned “Off”. Are you editing the color managed, or unmanaged data. Alternatively, even if color management is “On” in the workstation, you may want to verify that the appropriate profile is in use in both Acrobat and the editing application, otherwise your color adjustments will be off.

When Acrobat 4 was announced, it was hailed as ready for graphic arts applications because it now supported ICC color. Ironically, it was Acrobat 3 that was ready for graphic arts applications because it supported CMYK and spot colors.

Moreover, there is a wide spread misconception about what the ICC color support in Acrobat 4 is. Many people assume that you can attach an output device profile and everything will come out okay at the printer. If you were to do this, then you would be making your PDF document device dependent.

Adobe correctly did not adopt this approach. For the ICCBased color space, Acrobat supports the "A to B tags" (scanner and monitor profiles), and not the "B to A tags" (printer profiles). In other words, Acrobat is supporting the conversion of input data into common color space coordinates that can be understood by all downstream processes and printers. Because it is the responsibility of the output device to render the colors correctly, PDF documents remain portable.

Color output

Some of you might say that PostScript 3 RIPs can consume PDFs directly. This ignores certain realities, for example, that many PostScript 3 RIPs cannot and that the vast installed base of RIPs are PostScript Level 2 RIPs . Additionally, even though it is technically feasible to interpret and RIP PDFs directly, it is rarely done. The RIPs must also support PostScript for all the traditional workflows. After all, they are still called PostScript RIPs. So most often, even in a PostScript 3 RIP, PDFs are converted to PostScript and then interpreted. But frankly, it is not important if the PDF is comsumed natively or not , as long as you get the desired output and any file conversion is rapid.

Overprint is another color output issue. The issue arises because the implementations have varied widely among the various PostScript levels. Level 1 did not support in-RIP separation. So all overprints were initiated in a pre-separated workflow. Level 2 supported in-RIP separations. In this implementation, you could overprint spot on spot, spot on process, process on spot, but not process color on process color. Postscript 3 supports all of the above mentioned modes plus overprinting on composite devices with separate colorants, enabling some interesting proofing possibilities. Adobe cautions that overprint implementations are device dependent. How this behaves in your workflow with your devices, you will have to determine for yourself.

Adobe realized that PostScript will be used to print PDFs rich in graphic arts content. PostScript already has a mechanism for taking common device independent color coordinates and converting them to device color coordinates, that is, into the CMYK color gamut. It is called a Color Rendering Dictionary (CRD) and its inclusion in PostScript pre-dates ICC. Each PostScript RIP has default CRDs for its output device. So, to a first approximation, colors on a proof will be consistent with final output.

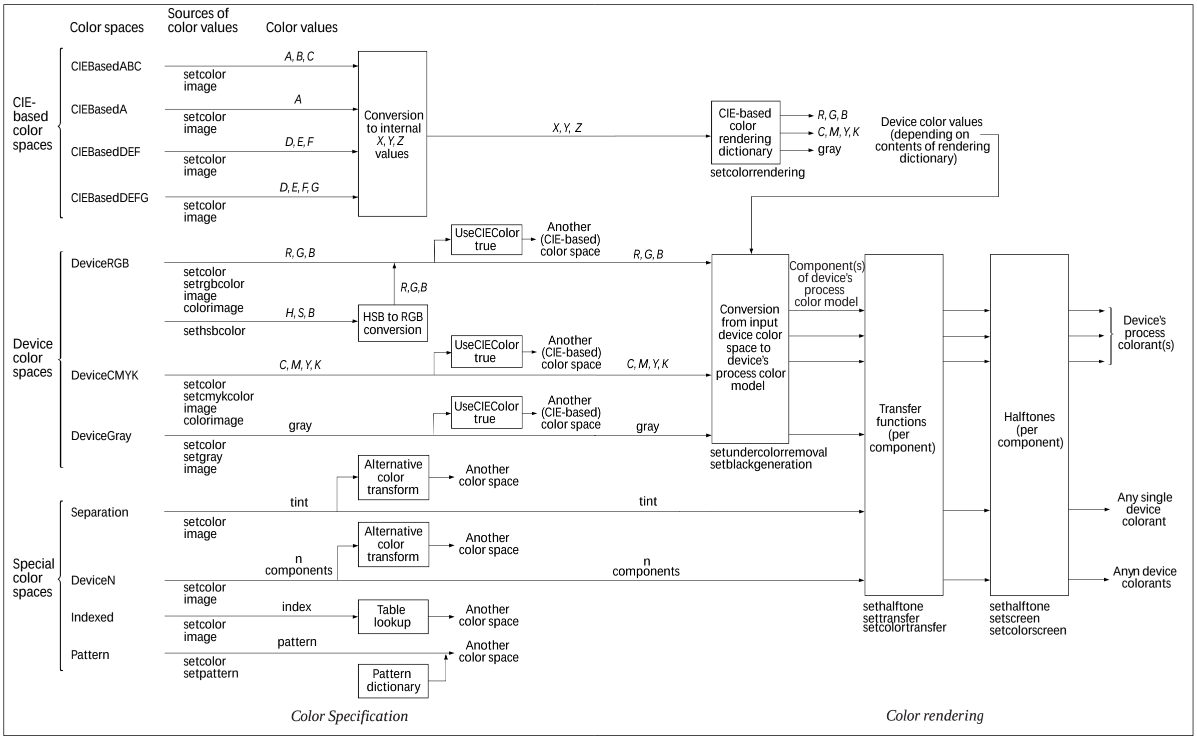
Ideally, you should have different CRDs for different combinations of paper, ink, and printing technology (ink jet vs. lithography vs. flexo, etc.). Theoretically, this is possible with PostScript and would give you the most consistent color reproduction regardless of final printing conditions. Adobe provides some help here. But not all applications take advantage of it. The printer drivers (see table below) can convert ICC printer profiles to CRDs by rendering intent, include them in the PostScript file and download them to the printer to be used for the current job. So indirectly, you could adjust CRDs by adjusting ICC profileswith a tool such as ProfileMaker from GretagMacbeth.

ICC Profile to CRD Initial Support			
Platform	**ICC system**	**Adobe driver**	**Non-Adobe driver**
Macintosh	ColorSync	AdobePS 8.4	Apple LaserWriter 8.4
Windows 95/98	ICM	AdobePS4	Microsoft PScript4
Windows NT4	ICM	None	None
Windows 2000	ICM2	AdobePS5	Microsoft PScript 5

© 2000 Lantana Research Software Corporation

As stated previously, ICC and PostScript recognize four rendering intents. In principle, there should be a CRD for each. But not all RIPs have all four CRDs. Some map the rendering intents into as few CRDs as one. You will want to pay attention to this if you are using device independent color and render intents are important to you.

Below is a diagram that shows how data defined in different color spaces traverse different paths as it is being processed in a PostScript RIP.


Color spaces
Sources of color values
Color values
CIE-based color spaces
CIEBasedABC
CIEBasedA
CIEBasedDEF
CIEBasedDEFG
setcolor image
A, B, C
A
D, E, F
D, E, F, G
Conversion to internal X, Y, Z values
X, Y, Z
CIE-based color rendering dictionary
setcolorrendering
R, G, B
C, M, Y, K
gray
Device color values (depending on contents of rendering dictionary)
Device color spaces
DeviceRGB
DeviceCMYK
DeviceGray
setcolor setrgbcolor image colorimage
sethsbcolor
setcolor setcmykcolor image colorimage
setcolor setgray image
H, S, B
HSB to RGB conversion
UseCIEColor true
Another (CIE-based) color space
Conversion from input device color space to device's process color model
setundercolorremoval setblackgeneration
Component(s) of device's process color model
Transfer functions (per component)
Halftones (per component)
Device's process colorant(s)
Special color spaces
Separation
DeviceN
Indexed
Pattern
tint
n components
index
pattern
setcolor setpattern
Alternative color transform
Another color space
Table lookup
Pattern dictionary
Any single device colorant
Anyn device colorants
sethalftone settransfer setcolortransfer
sethalftone setscreen setcolorscreen
Color Specification
Color rendering

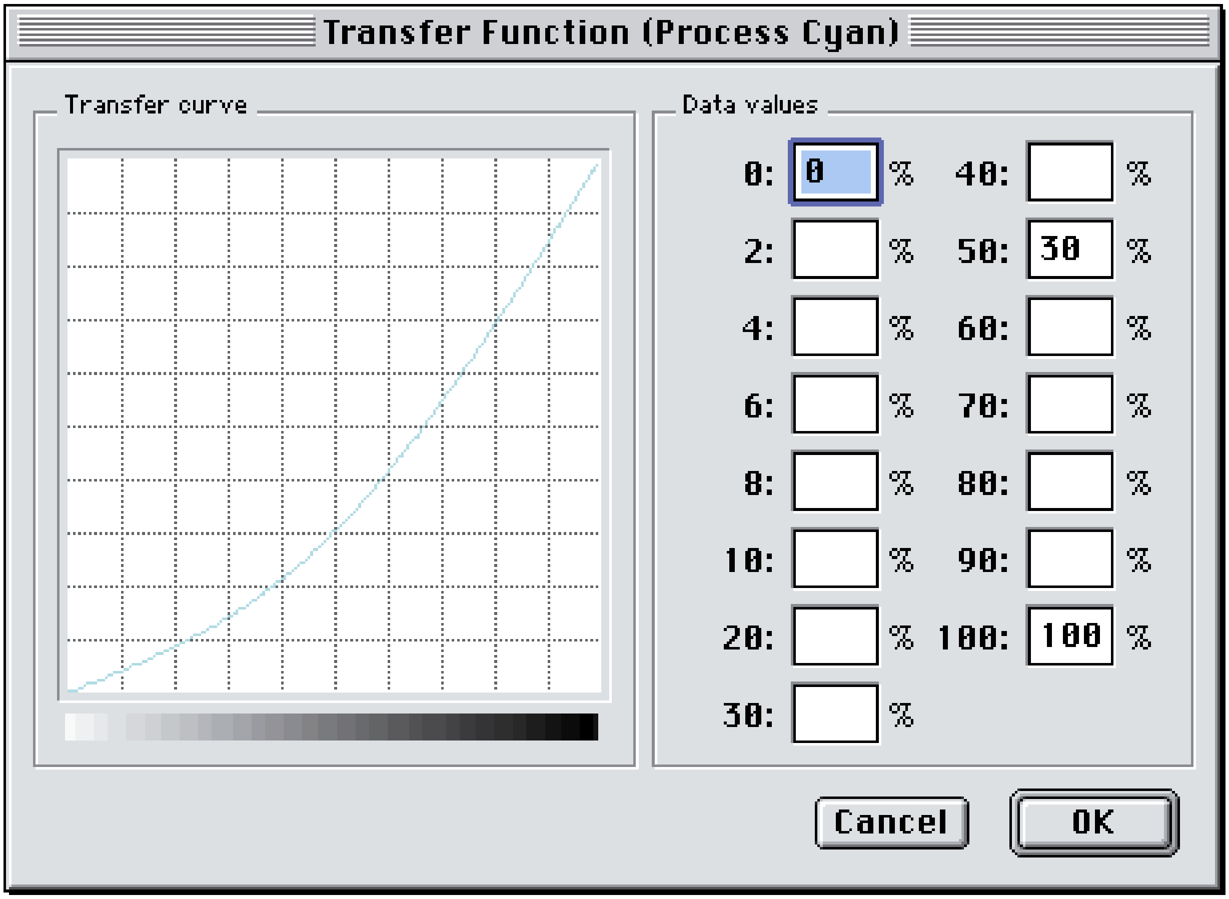
PostScript Color Processing

© 1999, Adobe Systems Incorporated

When is a color space not THE color space? Answer: when there is a "Default" color space defined in the RIP and UseCIEColor = true (see diagram). So, for example, the file calls for DeviceRGB, but a

DefaultRGB is defined in the RIP that could be a CIEBasedABC. This approach might also exist for DefaultGray and DefaultCMYK. This set of possibilities is included in the PDF/X-1 specification - a standard for using PDF for digital distribution of ads for magazines. Yes, this default color space can affect your output because it alters the path through which color data is processed in the RIP. But this slight of hand does not alter any of the other basic statements made in this article. Even though a "Default color space" is a technique for treating virtually all data in a ICC-based fashion, it has the disadvantage that it imposes an additional conversion and applies CRD processing to both the device independent and device dependent color coordinates, which may or may not be appropriate. Also, it can't be counted on as an approach that will exist consistently at all sites or on all printing devices.

From the chart you will notice that CIE-based data still passes through several device color processing steps before marks are placed on media. Indeed, all data is processed through these last few steps. This is the place where maximum control can be exerted in a consistent fashion. For this reason, when producing separations, Crackerjack has provided control of each ink's transfer function (shown below). This facility should be used primarily for dot gain compensation. But it is also a means of tweaking color on each print job without compromising your original PDF document.

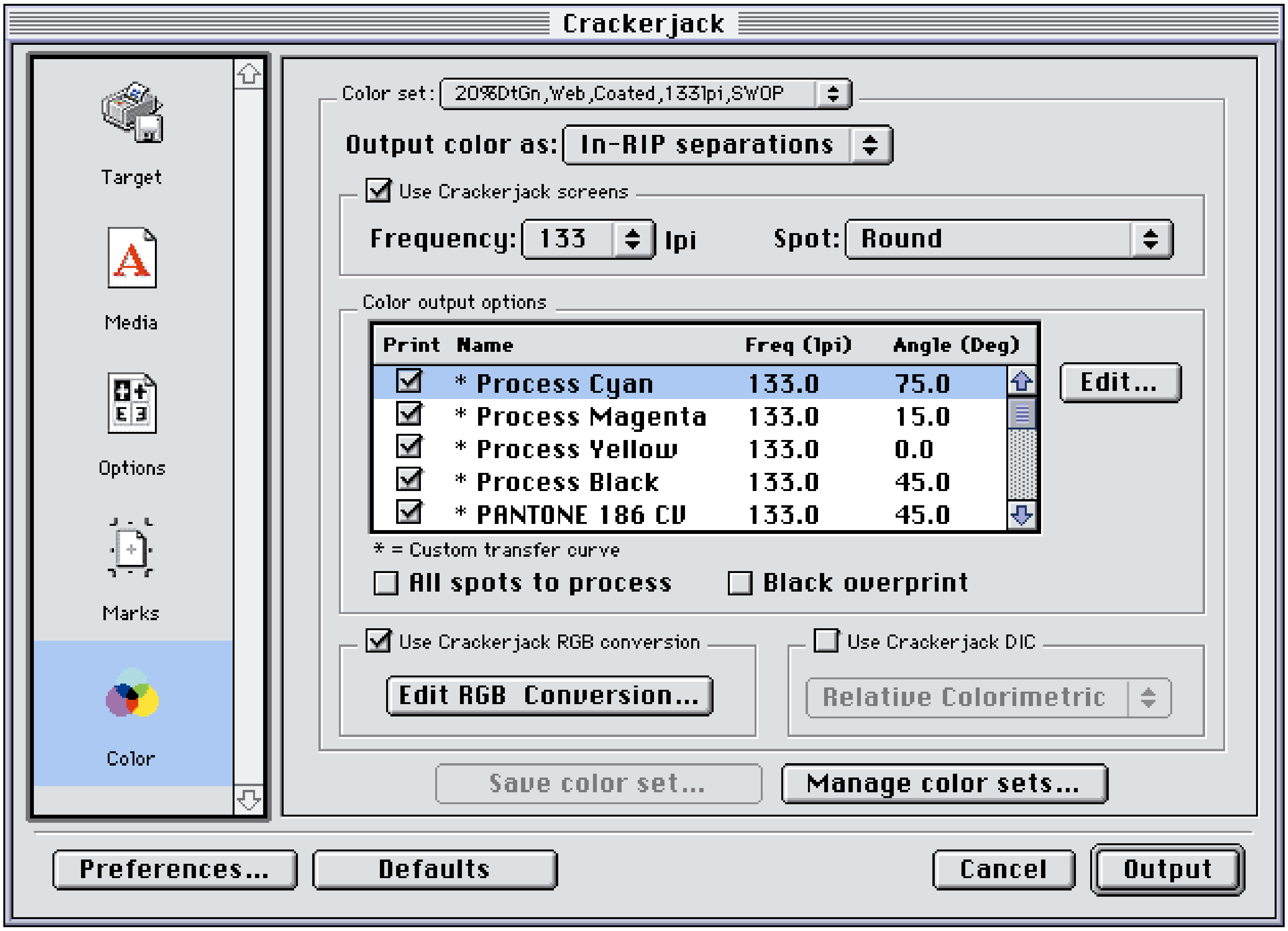

Transfer Function (Process Cyan)

Data values

0:	0 %	40:	%
2:	%	50:	30 %
4:	%	60:	%
6:	%	70:	%
8:	%	80:	%
10:	%	90:	%
20:	%	100:	100 %
30:	%		

Crackerjack Ink Transfer Function Panel

This diagram should also make us reconsider what it means to calibrate a printing system. There are three main data paths for color data through the RIP - CMYK, RGB, and CIE (ICC). A color management systems that claims to have an ICC output profile for your print conditions may not be addressing all three data paths. And typical "press fingerprinting" exercises only address the CMYK path. If you work exclusively in CMYK, then you only need to calibrate that path. But it is not uncommon to have two or all three of the basic color spaces together in the same document. For this reason Crackerjack allows one to save and recall “Color Sets”, i.e., all the parameters associated with all three data paths. An example follows.



Color set: 20%DtGn,Web,Coated,133lpi,SWOP

Output color as: In-RIP separations

☑ Use Crackerjack screens

Frequency: 133 lpi   Spot: Round

Color output options

Print	Name	Freq (lpi)	Angle (Deg)
☑	* Process Cyan	133.0	75.0
☑	* Process Magenta	133.0	15.0
☑	* Process Yellow	133.0	0.0
☑	* Process Black	133.0	45.0
☑	* PANTONE 186 CV	133.0	45.0

* = Custom transfer curve

□ All spots to process   □ Black overprint

☑ Use Crackerjack RGB conversion

□ Use Crackerjack DIC: Relative Colorimetric

Crackerjack Color Panel with “SWOP” Color Set

CMYK is the common data path for all PDF color data destined for print. Therefore, in all cases, this path should be calibrated first. Through the use of transfer functions and a test PDF document with known CMYK gradient percentages, this step should compensate for dot gain for the target paper, ink, and printing technology for the selected line screen frequency, spot function, and resolution. But also include 2 color overprints and 3 component gray scales to test color balance.

Independently, and assuming that they will be used, both the RGB and CIE data paths should be calibrated. The same PDF test files may be used, but specified in DeviceRGB and CalRGB or Lab, respectively. The RGB path can be used to set or verify the in-RIP UCR/black generation path to adjust how gray values are distributed between K and CMY. The effect on RGB text and images will be of greatest interest. The objective is to make sure your conversion mechanisms give you expected CMYK values. If interested, one could also generate a test file in CalRGB or Lab with the entire range of color values, which could be used to verify gamut mapping for each rendering intent.

Conclusion

The primary decision for you to make is whether to institute a CMYK or a color managed workflow. To the extent possible, life will be a lot easier if you can stick to one or the other 100% of the time. This is much easier if the original files are created internally. But if, you have multiple external sources for input files, then I recommend that you preflight them and convert them to your internal standards immediately. A color managed ICC workflow holds the promise of more reliable reproducible color, even at remote sites. A CMYK workflow is easier to manage and for your employees to understand and diagnose problems.

You can get excellent color out of the PDF workflows described above. You can have everything from duotones to spot colors. You can repurpose documents from print to web. The good news is that you have lots of choices. It is up to you.