Printer Systems-Based Separations

Adobe Developers Association

9 October 1997

Technical Note #5606
LanguageLevel 3
# Contents

1. Printer Systems-Based Separations 7  
   - Overview of Separations 7  
   - Overview of Printer Systems-Based Separations 8  
   - The Benefits of Printer Systems-Based Separations 9  

2. Additions and Changes to the PostScript Language 10  
   - Specific Changes to the Page Device Dictionary for Separations 10  
   - General Changes to the Page Device Dictionary 12  
   - General Changes to the Indexed Color Space 13  
   - General Changes to the Separation Color Space 13  

3. Printing Separations 14  
   - Page Device Parameters for Separations 15  
   - Error Conditions when Producing Separations 16  
   - Device-Specific Considerations for Separations Output 17  

4. Printing Composite Output on a Device that Supports Separations 18  
   - Page Device Parameters for Composite Output 18  
   - Error Conditions when Producing Composite Output 19  
   - Device-Specific Considerations for Composite Output 19  

5. Printing on Devices that Support only Composite Output 20  

6. Printer Systems-Based Separations and the DeviceN ProcessColorModel 21  

7. Tips and Techniques 23  
   - MaxSeparations 23  
   - Separations and EPS Files 23  
   - DeviceN and Separations 24  

8. Issues to Consider with using Printer Systems-Based Separations 25  
   - Issues with Page Device Parameters in Combination 25  
   - Issues with using the DeviceN Process Color Model with Separations 25
Preface

This Document

This is the original release for Printer Systems-Based Separations, a document that provides a detailed description of the LanguageLevel 3 extensions support printer systems-based separations on all PostScript® printers.

Intended Audience

This document is written for software developers who are interested in learning about printer systems-based separations, or taking advantage of these capabilities in an application that produces output for PostScript printing devices.

It is assumed that the developer is already familiar with how separations work in previous levels of the PostScript language.

Organization of This Document

Section 1, “Printer Systems-Based Separations,” gives an overview of separations in general, some definitions of composite versus separated output, and an introduction to printer systems-based separations.

Section 2, “Additions and Changes to the PostScript Language,” describes the language changes, mostly in the page device dictionary, for printer systems-based separations. This section also discusses changes to the indexed Separation color spaces and mention is made of the new color space and process color model called DeviceN.

Section 3, “Printing Separations,” discusses the general method for printing separations on a PostScript device that supports them.

Section 4, “Printing Composite Output on a Device that Supports Separations,” describes the general method for printing composite output on a separations device.
Section 5, “Printing on Devices that Support only Composite Output,” gives some general information on composite-only output devices.

Section 6, “Printer Systems-Based Separations and the DeviceN ProcessColorModel,” details some of the requirements and issues surrounding the use of the DeviceN process color model.

Section 7, “Tips and Techniques,” covers some tips and techniques for producing separations.

Section 8, “Issues to Consider with using Printer Systems-Based Separations,” discusses some of the issues, limitations, and/or caveats with producing printer systems-based separations.

Related Publications


Statement of Liability

THIS PUBLICATION AND THE INFORMATION HEREIN IS FURNISHED AS IS, IS SUBJECT TO CHANGE WITHOUT NOTICE, AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY ADOBE SYSTEMS INCORPORATED. ADOBE SYSTEMS INCORPORATED ASSUMES NO RESPONSIBILITY OR LIABILITY FOR ANY ERRORS OR INACCURACIES, MAKES NO WARRANTIES OF ANY KIND (EXPRESS, IMPLIED, OR STATUTORY) WITH RESPECT TO THIS PUBLICATION, AND EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSES, AND NONINFRINGEMENT OF THIRD-PARTY RIGHTS.
Printer Systems-Based Separations

1 Printer Systems-Based Separations

1.1 Overview of Separations

PostScript color output devices produce full color output by combining their process colors (or colorants) in varying amounts to create the final composite page (piece of media) or pages. In the case of monochrome devices, the one process color is black, and the final composite output is creating by using black in varying amounts. Most output devices produce a single composite page on which all process colors have been combined. That is, one composite page is printed for each document (or virtual) page sent to the output device (this may not reflect the actual number of physical pages printed if the number of copies is set to a value greater than 1). There are other types of output devices, such as imagesetters, that produce a collection of pages per document (or virtual) page, the collection being one page for each process color. Each of these pages is called a separation. Each separated page or piece of media is a monochromatic (usually black) representation of a single process color. When these separated pages are later combined (as on a printing press) and the correct amounts of device colors/inks are applied to them, a full-color, composite page results.

The terms separation and separations are often misused. In the past, they have been used to refer to any and all of the following:

- The data describing one plane of color.
- The planes of data used to print separations.
- The actual colorants of the device.
- The Separation color space.
- The SeparationColorNames key in the page device dictionary.
In the context of this document, a PostScript printer that produces separations generates \( n \) pages (or pieces) of media (such as paper or film). These pieces of media are known as separations. The *colorants* (the process colors) of a PostScript printer should not be confused with its separations. For example, given a CMYK device with a `DeviceCMYK ProcessColorModel`, four pages can printed, one for each of the Cyan, Magenta, Yellow, and Black separations. The colorants of the device are CMYK. A specific device colorant (such as Cyan) should be referred to as a separation only if the device is generating separations, where one of the separations (namely, Cyan) corresponds to the colorant.

1.2 **Overview of Printer Systems-Based Separations**

In previous levels of the PostScript language, only imagesetters could generate separations as individual pages. Other printing devices depended on the host computer to produce the imaged data for each separation. To print separations on a device required the host-based application to organize the PostScript job by named colorants and execute the `setcolorspace` and `showpage` operators for each separation. This was an intrinsically unreliable process, especially when applied to encapsulated PostScript (EPS) files.

Printer systems-based separations, also known as *selectable separations*, is a LanguageLevel 3 feature that enables all printing systems (both monochrome and color) to support requests for separations, where a separation is printed page that corresponds to one color plane of data (see the previous section). This new feature of the PostScript language enables all PostScript printers to generate data for one or more color plane separations defined in the job instead of requiring that separate PostScript files be generated by the host for each color plane. For example, using printer systems-based separations, a monochrome printer can print the Cyan plane of a CMYK job if requested. Separations are printed in a monochrome color, which will typically be black for the best contrast.

In previous levels of the PostScript language, pre-separated jobs needed to be preprocessed in order to print as *composite* color, where composite color is the color that results from combining all color planes or colorants. Since all PostScript printers that support the LanguageLevel 3 feature printer-systems based separations can create separations, PostScript jobs no longer need to be pre-separated. This enables color proofing devices to accept a PostScript job produced for an imagesetter, but produce composite color output when requested.
1.3 The Benefits of Printer Systems-Based Separations

There are several benefits in using printer systems-based separations:

- Valuable to applications developers/users who want to produce and view separations for their print jobs. The application has a level of control, through the PostScript language, to specify colorants to be printed (separate or together), regardless of the `ProcessColorModel` in effect or the actual inks of a specific PostScript printer.

- Enables applications producing the data for the separations to produce better PostScript files, which will result in more accurate output. Part of this is the result of eliminating the need for extra code to produce host-based separations, allowing the PostScript printer to produce the separations, instead.

- Applications users have the ability to emulate what can be done on a high-end separations device such as an imagesetter.

- Proofing can be performed on any PostScript 3 printer.

- Application users can use a composite workflow throughout because the same file can be used for both composite and separated output. When separations are needed on the file, they are generated in the printer, no on the host.

- Printer systems-based separations support all color spaces and color models, including the new `LanguageLevel 3` color space `DeviceN`.

- It is now much easier to separate device-independent color (DIC). This is very difficult and impractical to do with host-based separations.

*Note*  For more information on the `DeviceN` color space and/or the `DeviceN` process color model, see Technical Note #5604, “`DeviceN Color Space and Color Model,” available from the Adobe Developers Association.
2 Additions and Changes to the PostScript Language

This section covers all of the significant changes and/or extensions to the PostScript language to support printer systems-based separations. In addition, other, more general, changes have been made to existing language elements so that they may be used with separations.

One new page device parameter called MaxSeparations has been added in LanguageLevel 3. In addition, general modifications have been made to the previously introduced parameters ProcessColorModel, Separations, and SeparationColorNames. Finally, specific changes have been made to the parameter SeparationOrder to support printer systems-based separations. All of these additions and changes are covered in Sections 2.1 and 2.2.

The Indexed and Separation color spaces have been modified in LanguageLevel 3. Their use with printer systems-based separations is covered in Sections 2.3 and 2.4.

A new color space has been introduced in LanguageLevel 3, called DeviceN, which allows for the specification of other than the standard three or four colors. DeviceN is also a valid value for ProcessColorModel. Issues surrounding the use of the DeviceN ProcessColorModel are covered at the end of this document in Section 6.

2.1 Specific Changes to the Page Device Dictionary for Separations

MaxSeparations Parameter

The new parameter MaxSeparations returns an integer value that specifies the number of separations that the current device can produce, regardless of whether or not that device will be producing separations. This is a read-only parameter whose returned value is in the range 1 to 250.

The number of separations a PostScript interpreter can produce depends on the amount of memory allotted for separations. The PostScript interpreter on the device sets this value after all other separations parameters, and other device parameters, have been set (such as PageSize, Resolution, ProcessColorModel, and DeviceRenderingInfo). The PostScript job must conform to the capabilities of the current output device (as defined by MaxSeparations); otherwise, page device policy will be consulted and the output may not be as expected (depending on the current policy).

Application developers can query this read-only value on each device and use the returned value as an upper limit on the number of elements in the SeparationOrder array.
Separations Parameter

The `Separations` parameter specifies whether or not to produce separations on the PostScript printer. If the value of `Separations` is true, the device will produce each page as an individual separation – one separation for each rendering colorant. If the value is false, the device will produce a single composite page.

SeparationOrder Parameter

The `SeparationOrder` parameter is an array whose value specifies the colorants to be produced by the device. This array can contain names or strings, or some combination of both. Legal entries in the array are the current process color names and any additional names specified by `SeparationColorNames`. If the array is empty, the PostScript interpreter determines the order of the separations it will produce, based on the current value of `ProcessColorModel` and `SeparationColorNames`. The array does not have to contain the name of every colorant implied by `ProcessColorModel` and given explicitly by `SeparationColorNames`. The array can, instead, contain a selected subset of the rendering colorants available.

If the value of the `Separations` parameter is true, a separation will be produced for each occurrence of a colorant name; multiple occurrences will produce multiple separations. Only the colorants named in the `SeparationOrder` array will be produced although all named separation colorants are defined (instead of reverting to the alternate color space). Each separation will be produced in the order specified in the `SeparationOrder` array.

If the value of `Separations` is false, `SeparationOrder` specifies which colorants are to be applied to a composite page; in this case, the order of entries in the array is not important. The size of `SeparationOrder` is also not important; it can have as desired for composite color because there is no memory constraints.

Note  For more information on the `MaxSeparations`, `Separations`, and `SeparationOrder` parameters of the page device dictionary, see Table 4.17 of the Supplement: PostScript Language Reference Manual.
2.2 General Changes to the Page Device Dictionary

ProcessColorModel Parameter

The ProcessColorModel parameter has been extended to support the DeviceN color model. This parameter specifies the colorant model used for rendering process colors in the device. It affects rendering for all color spaces, with the exception of Separation and DeviceN color spaces. These two color spaces produce colorants only if the alternate color space is not invoked; they must explicitly define colorants to be marked. When printing composite pages, the Separation and DeviceN color spaces only produce expected results when the colorants specified by ProcessColorModel match those specified by Separation or DeviceN. Otherwise, the alternative color spaces is invoked, in which case, the alternative color space cannot be DeviceN. When producing separations, colorants associated with the Separation and DeviceN color spaces should always be honored, assuming that they have been requested properly through the SeparationColorNames and SeparationOrder parameters. The ProcessColorModel parameter does not affect the interpretation of color values in any color space; it only controls the rendering method.

The legal values for ProcessColorModel are /DeviceGray, /DeviceRGB, /DeviceCMYK, /DeviceCMY, /DeviceRGBK, and /DeviceN (alternately, these could be specified by strings rather than names). The rendering colorants for each of the process color models are as follows:

- DeviceGray: /Black
- DeviceRGB: /Red, /Green, /Blue
- DeviceCMYK: /Cyan, /Magenta, /Yellow, /Black
- DeviceCMY: /Cyan, /Magenta, /Yellow
- DeviceRGBK: /Red, /Green, /Blue, /Black
- /DeviceN: (there are no implied colorants)

Valid colorants for the DeviceN ProcessColorModel are device-specific; the DeviceN entry in the OutputDevice resource will provide the set of colorants for a specific device.

The process colorant names (such as /Red /Green /Blue) specified by the process color model (/DeviceRGB) are used to select halftones in a Type 5 halftone dictionary; the names are implicitly included in SeparationColorNames.
**SeparationColorNames Parameter**

The `SeparationColorNames` parameter has been extended to support printer systems-based separations and the `DeviceN` color model. `SeparationColorNames` and `ProcessColorModel` together specify all colorants of a PostScript job. Process colors are implicitly included in this array (in other words, process colors like Red, Green, and Blue, or Cyan, Magenta, Yellow, and Black, do not have to be specified as part of this array). Therefore, expect for `DeviceN`, all values of `ProcessColorModel` have colorants that do not need to be explicitly stated in the `SeparationColorNames` array. For the `DeviceN` color model, all device colorants must be specified explicitly in the array, since there are no pre-defined process colorants.

Entries in the array can be either names or strings, or some combination of both. Duplicate entries in the array are ignored, and the order of the array entries is not important.

*Note* For more information on the `SeparationColorNames` or the `ProcessColorModel` parameter of the page device dictionary, see Table 4.17 of the Supplement: PostScript Language Reference Manual.

### 2.3 General Changes to the Indexed Color Space

The `Indexed` color space has been modified to allow the base color space to include the `Separation` and `DeviceN` color spaces. The main reason for this change is to give applications a convenient method for representing spot and multi-component colorized gray images as composite images for printer systems-based separations.

*Note* Using the `Separation` color space as the base color space for the `Indexed` color space will generate an error on previous levels of the PostScript language.

### 2.4 General Changes to the Separation Color Space

The `Separation` color space is not a separation by definition. This color space specifies a named color and an alternate color space to use if the color cannot be printed by the PostScript printer.

In general, the named color (such as `/SkyBlue`) is converted to a rendering colorants using the alternate color space. In the case where the name of the color space matches a rendering colorants (such as `/Cyan`), the color space directly marks, without color conversions, using that rendering colorant.
3 Printing Separations

Since the PostScript job designates the process color model and the named colorants, the PostScript interpreter must support all conversions of any standard color model. Therefore, all PostScript 3 printers are able to handle each of those standard color models. This covers the case when a PostScript job may request a `ProcessColorModel` value other than `DeviceGray` when `Separations` is true. For example, if the `ProcessColorModel` value is set to `DeviceCMYK` when `Separations` is true, the monochrome device will render color objects into the CMYK colorants and produce separations for each of these colorants. Further, any color rendering dictionary (CRD) that is used by other color models are also provided, even if only an approximation is created for printing separations. In other words, CRDs are used during the conversion to device spaces when doing separations in the same way that they are used when doing composite pages. The CRDs involved are the ones that are current in the graphics state, whether provided by the device or by the current PostScript job.

There are four page device parameters that are consulted when `Separations` is true. These are `MaxSeparations`, `ProcessColorModel`, `SeparationColorNames`, and `SeparationOrder`. These four parameters determine how best to address the color space requests of named colors and which separations are to be produced. They are also consulted when `Separations` is false. The values of `ProcessColorModel`, `SeparationColorNames` and `SeparationOrder` must be consistent when `Separations` is false.

The number of separations that may be produced on a particular PostScript printer can vary, especially when there is insufficient memory. An error could result whenever the number of separations explicitly declared in `SeparationOrder` or implied in `SeparationColorNames` cannot be generated (page device policy will be consulted in this case). The `MaxSeparations` parameter is provided so that the appropriate size of the `SeparationOrder` array can be determined by the job. PostScript printers that provide composite output of more than one toner or ink can often provide the same number of separations as there are toners or inks (that is, `MaxSeparations` may be greater than 1 when the device itself has multiple toners/inks used when printing).

An application can check the value of `MaxSeparations` by either directly polling the printer or checking its PostScript printer description (PPD) file. In the case of using the PPD file, it must be remembered that users of the PostScript printer can make changes to the device that would not necessarily be reflected by the PPD.
Note  If a configuration error is raised, the PostScript job producer (application that produced the PostScript code) should reduce the size of the SeparationOrder array to conform to the storage capacity of the device, given that one separation is guaranteed for every PostScript printer.

Each colorant of the SeparationOrder array will be output regardless of whether or not there are marks on the page for that rendering colorant. In other words, a page of output will be produced for each colorant named in the array even if that colorant does not contribute to the information on the page. It should be a rare occurrence, though, that a completely blank sheet of media will be produced for any one colorant. Typically, a PostScript job itself may mark or identify the name of the colorant (in that colorant), although this is totally up to the application that generated the PostScript job.

Note  For more information on page device policies, see Section 4.6 and Appendix D of the Supplement: PostScript Language Reference Manual.

3.1 Page Device Parameters for Separations

The Separations parameter exists in the page device dictionary in PostScript 3 devices, and in some older PostScript printers. If the parameter does not exist, the device is incapable of producing separations. Having the Separations parameter implies that ProcessColorModel may be set to values other than what the device will allow for composite output. In other words, if the Separations parameter exists and its value is true, the current value of ProcessColorModel need not reflect the actual toners or inks provided by the PostScript printer. The SeparationOrder parameter is consulted to determine which separations will be produced and output to the device. Each separation will print in monochrome (black and white), but will represent the specific colorant. If the Separations parameter exists, but its value is false, the request may be rejected.

An empty SeparationColorNames array implies that only process colors are involved.

SeparationOrder may explicitly specify a colorant more than once and pages produced will reflect each entry in the array.
The following is a sample call to the `setpagedevice` operator to produce two separations. This code assumes that the value of `MaxSeparations` is at least 2.

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

The values in the `SeparationOrder` array can only be `/Cyan`, `/Magenta`, `/Yellow`, or `/Black` since `ProcessColorModel` is `DeviceCMYK` and the `SeparationColorNames` array is empty. If the value of `MaxSeparations` was greater than or equal to 4 and the `SeparationOrder` array was empty, then all four colors would be printed as separations. If the value of `MaxSeparations` was less than 4 and the `SeparationOrder` array was empty, then a configuration error would result.

The next example shows a sample call to the `setpagedevice` operator to produce three separations. This code assumes that the value of `MaxSeparations` is at least 3.

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

3.2 Error Conditions when Producing Separations

Page device policy will be consulted in the following instances, after the request for separations has been made:

- The number of unique separations requested, as defined by `SeparationOrder`, cannot be satisfied. The recommended policy for `SeparationOrder` is 0, which means generate a configuration error. If the value of `MaxSeparations` is less than 250, the device must have a memory constraint and the `SeparationOrder` array must be modified to conform to the number of separations the device may produce successfully. If the policy is 1 (which means ignore the request), the output behavior in this case is product-specific.

- The entries in `SeparationOrder` do not reflect available colorants as specified by `SeparationColorNames` and `ProcessColorModel`. 
• The page device parameters **Separations** and **Duplex** are both true. If the policy for **Duplex** is not 0, then the request for separations will proceed as simplex (single-sided) output.

• **MirrorPrint** or **NegativePrint** requests are made. Applications that produce separations may expect a device to provide for mirror and negative printing. If either parameter is not defined for a PostScript printer, **PolicyNotFound** will be used (which usually means that the request will be ignored). Applications that perform their own mirroring or negative printing are not limited by this policy.

3.3 Device-Specific Considerations for Separations Output

The PostScript interpreter sets a value of **MaxSeparations** that is consistent with current page device parameter settings. In most cases, the number of colors in the composite image dictates the number of separations. The amount of memory set aside for separations may limit the number of separations to less than the number of colors in the composite image. However, options such as an internal or external hard disk can remove any limit to the number of separations possible. The application can also send the file once for each of the colorants named in the **SeparationOrder** array.

A **MaxSeparations** value of less than 250 indicates some type of memory constraint. The PostScript job must specify a **SeparationOrder** array to match the value of **MaxSeparations** for the expected output (separations) to be produced. A **SeparationOrder** array of one element will always succeed.
4 Printing Composite Output on a Device that Supports Separations

Composite output is defined as a single output page per document page, using the colors/inks of the PostScript printer. The list of allowed values for ProcessColorModel for producing composite output is provided in the OutputDevice resource. If there is only one color model available, or if the choice of color models is not selectable through software, the value of ProcessColorModel when the value of the Separations parameter is false is the color model used for composite images.

The ProcessColorModel, SeparationColorNames, and SeparationOrder parameters specify the colorants to be used for the composite image. The alternative color space (for the Separation and DeviceN color spaces) will be used when any color plane is not defined by either ProcessColorModel or SeparationColorNames. SeparationOrder may specify a subset of colorants to use for the composite image. Duplicate entries may be specified in SeparationOrder, but it is only applied once on the output.

If a colorant does not correspond to an ink of the device, it will be emulated using the available inks during rendering.

4.1 Page Device Parameters for Composite Output

The value of ProcessColorModel should be consistent with the capabilities of the specified PostScript printer. The SeparationOrder parameter is consulted to determine which colorants will be produced for composite output. Only the unique colorants specified by SeparationOrder will be produced; duplicates will be ignored.

The following is a sample call to the setpagedevice operator to produce composite output from two of the four colorants specified in ProcessColorModel.

```
<<
/Separations false
/ProcessColorModel /DeviceCMYK
/SeparationColorNames []
/SeparationOrder [/Cyan /Yellow]
>> setpagedevice
```

where /Cyan and /Yellow are the names of the colors selected to print. They can only be /Cyan, /Magenta, /Yellow, or /Black since ProcessColorModel is DeviceCMYK. In this example, the colorants of the PostScript job are limited to the current value of ProcessColorModel because the SeparationColorNames array is empty. Thus, SeparationOrder must only name colorants of the current ProcessColorModel. It is worth mentioning that if the ProcessColorModel request had been rejected (on a monochrome
device, for example). SeparationOrder would also have been rejected because it was inconsistent with the implied colorants of ProcessColorModel. If the policy for each of these parameters was to ignore the request, composite output would have been generated with all color values rendered and output as gray.

4.2 Error Conditions when Producing Composite Output

Page device policy will be consulted in the following instances:

- The value of Separations is false and the value of ProcessColorModel is inconsistent with the capabilities of the device. In other words, the requested value of the ProcessColorModel, when Separations is false, will only be accepted when that ProcessColorModel value is one that the device can support; rejection of the request results in no change to the ProcessColorModel. The recommended policy for ProcessColorModel is 1, which means ignore the request.

- The value of SeparationColorNames does not specify the colorants of the DeviceN color model of the output device.

A configuration error will result if the SeparationOrder array specifies colorants not identified by either SeparationColorNames or the value of the ProcessColorModel currently in effect.

4.3 Device-Specific Considerations for Composite Output

The PostScript printer should be able to handle composite printing when a subset of its color planes are selected in the SeparationOrder array.

The PostScript interpreter determines how colorants in SeparationColorNames are to be rendered when they do not reflect the colors/inks of the device. The device may convert such colorants through device-specific code, although few devices will do this.

The device supplies (default) values of ProcessColorModel and SeparationColorNames when the value of the Separations parameter is false in order to have currentpagedevice reflect the capabilities of the device (before any requests are made by the PostScript job). SeparationColorNames would be an empty array unless there are spot colors and/or the value of the ProcessColorModel is DeviceN.

The allowed values of ProcessColorModel for composite output should be specified in the OutputDevice resource.

Note For more information on printing composite output when the ProcessColorModel is DeviceN, see Section 6 of this document.
5 Printing on Devices that Support only Composite Output

For all LanguageLevel 1 and most LanguageLevel 2 PostScript printers, separations are not supported; that is, these devices only produce composite output. For these devices, the Separations and MaxSeparations parameters do not exist in the page device dictionary. There are some LanguageLevel 2 PostScript printers that support separations through the Separations parameter (such as imagesetters), but they do not support the MaxSeparations parameter.

The PolicyNotFound key of the Policy subdictionary within the page device dictionary will be consulted if the Separations and MaxSeparations parameters are not present in a setpagedevice request of a PostScript job being sent to a composite-only device. If the current policy is to ignore the request, composite output will result for the ProcessColorModel consistent with the device. For those named colors not in the ProcessColorModel and SeparationColorNames array, the alternative color space declaration will be used to convert to process colors of the device.

Note All PostScript 3 interpreters support separations through the Separations and MaxSeparations parameters in the page device dictionary.

Note For more information on printing composite output when the ProcessColorModel is DeviceN, see Section 6 of this document.
Printer Systems-Based Separations and the DeviceN ProcessColorModel

Given a device that supports the DeviceN process color model, a PostScript job may request that the device behave as a DeviceN device with a certain set of colorants as specified in SeparationColorNames. In such cases, a product-dependent decision is made at setpagedevice time as to whether or not the device can behave as the requested DeviceN device. A request for the ProcessColorModel value of DeviceN will only be accepted when the device has the means to convert standard color spaces, such as DeviceRGB and DeviceCMYK, into the set of DeviceN colorants specified by SeparationColorNames.

If the request for ProcessColorModel as DeviceN is rejected (because the device has no such capability), some other ProcessColorModel value will be active (such as the previous or default model) and rendering will be into its colorants (DeviceGray for monochrome) unless there is a setcolorspar call that explicitly marks in colorants identified by SeparationColorNames. There is no pre-specified method of converting incoming data defined as DeviceGray, DeviceRGB, or DeviceCMYK into DeviceN colorants. PostScript device-specific color conversions are supplied for these cases.

The value of ProcessColorModel should be consistent with the capabilities of the specified PostScript printer. When the value of ProcessColorModel is DeviceN, the SeparationColorNames parameter must be used to identify colorants, at a minimum, which make up the DeviceN color model of the output device.

Printing Spot Colors

A PostScript printer that supports spot colors may be configured for the DeviceN (or any other) process color model. This may be a rare occurrence since it is more common to add spot colors with DeviceCMYK. If the PostScript printer is configured for DeviceN, though, the device is responsible for the method of converting device color spaces, such as DeviceRGB or DeviceCMYK, to its DeviceN colorants. The methods used to perform the conversions and the final choice of colors/inks are device-specific.

A PostScript job can ensure imaging into spot colors by using Separation and DeviceN color spaces explicitly.
Printing Composite Output

A product that supports the ProcessColorModel of DeviceN must be able to convert standard color space references (as allowed for the alternative color space) to the color planes of the output device used in composite images. Page device policy will be consulted if SeparationColorNames only specifies a subset of the colors available on the device. The output device may offer color space conversion routines for varying sets of its colors/inks to allow for all combinations possible, whereby a subset of colors would be allowed (and policy is not consulted).

Printing on Composite-Only Devices

Whenever the ProcessColorModel is DeviceN, the page device policy will be consulted if the colorants named in SeparationColorNames is inconsistent with the actual colors of the device.
7 Tips and Techniques

7.1 MaxSeparations

Applications that support separations will most likely be setting them up after consulting the PPD file or communicating with the PostScript printer directly to get the value of MaxSeparations. This value can be used to verify that the number of entries in the SeparationOrder array is less than or equal to MaxSeparations; that value can also be used to define SeparationOrder with the exact number of separations that the current PostScript printer can support.

It is entirely possible that the number of separations requested is greater than the number allowed by MaxSeparations. At this point the number of elements in the SeparationOrder array would have to be changed to reflect this upper limit if it is desired that the job create the largest allowable number of separations. Alternatively, a job can be sent multiple times with different values of the SeparationOrder array to get around the limitation of the MaxSeparations parameter. For example, if four separations are needed and the value of MaxSeparations is 2, then two SeparationOrder arrays can be built and the job sent twice, once with each array. This technique also replaces the older method of redefining PostScript operators, as described in Technical Note #5044, “Color Separation Conventions for PostScript Language Programs”.

7.2 Separations and EPS Files

Applications are currently available that can perform separations on PostScript jobs that contain EPS. Some of these applications perform the separations on the host, others perform the separations in the Raster Image Processor (RIP). In either case, the process for creating the separated EPS files can be limited or result in inefficient PostScript code. Separations in the RIP is the preferred method of the two since there is less work involved in getting the PostScript interpreter to generate separations. Even so, applications that produce separations in the RIP can still have difficulty handling EPS from various sources.

Since all PostScript 3 printers can create separations, it becomes much easier and more straightforward to define and use separations with our without EPS.

If the color space specified in the EPS code is unknown (cannot be determined), the resulting separations could be unpredictable.
7.3 DeviceN and Separations

It is not recommended that DeviceN be used as the current process color model for printer systems-based separations because very few devices will initially support the DeviceN process color model. Additionally, devices that support the DeviceN process color model and have the same set of colorants may still not generate the same separated output because of their device-specific color conversions. DeviceN can still be used, though, as a color specification for a job that will be separated. DeviceN color specification gives the PostScript job total control over marks made in named colorants.
8 Issues to Consider with using Printer Systems-Based Separations

8.1 Issues with Page Device Parameters in Combination

Some combinations of page device parameters will require that page device policy be consulted. The following is a list of some combinations and their likely outcomes:

- If the values of the Duplex and Separations parameters are both true, the device will produce simplex separations (one-sided output).

- If the device is CMY (has only CMY inks and no black ink), the color of the separation output may not necessarily be black.

- If the PostScript job makes requests for actions performed at page boundaries, such as jog, these actions will go into effect after each separation (or sheet of media) is produced, not for each logical or virtual page of the job.

8.2 Issues with using the DeviceN Process Color Model with Separations

Since the DeviceN process color model will most likely be available on only a few PostScript printers, it is not recommend that it be used when separations are being made (Separations is true). Use the DeviceN process color model for only very device-specific jobs. DeviceN can be used for color specification when creating separations on any PostScript 3 printer, by properly specifying the elements of SeparationColorNames. For example, if it is desired to create separations for six colors – such as /Cyan, /Magenta, /Yellow, /Black, /Orange, /Green – as along as the names in SeparationColorNames matches the names used in the DeviceN color specification, separations should be made.
Appendix A

Bibliography of Outside Sources

While this is not an exhaustive list of references, it will give the reader some sources for the concepts covered in this document.

For more information on the DeviceN color space or the DeviceN process color model, see Technical Note #5604, “DeviceN Color Space and Color Model.” This Technical Note is provided by the Adobe Developers Association.

For more information on the LanguageLevel 2 approach to separations see Technical Note #5044, “Color Separation Conventions for PostScript Language Programs”. This Technical Note is also provided by the Adobe Developers Association.
Index

C
currentpagedevice 19

D
DeviceCMYK 14, 16, 18, 21
DeviceGray 14, 21
DeviceN v, vi, 9, 10, 12, 13, 18, 19, 21, 24, 25
DeviceRGB 21
Duplex 17, 25

I
Indexed v, 10, 13

L
LanguageLevel 1 20
LanguageLevel 2 20, 27
LanguageLevel 3 v, vi, 8, 9, 10

M
MaxSeparations 10, 11, 14, 16, 17, 20, 23
MirrorPrint 17

N
NegativePrint 17

O
OutputDevice 12, 18, 19

P
Policy 20
PolicyNotFound 20

ProcessColorModel 8, 9, 10, 12, 13, 14, 15, 16, 18, 19, 21, 22

S
Separation v, 10, 12, 13, 18, 21
SeparationColorNames 7, 10, 11, 12, 13, 14, 16, 18, 21, 22, 25
SeparationOrder 10, 11, 12, 14, 15, 16, 17, 18, 19, 23
Separations 10, 11, 14, 15, 17, 18, 19, 20, 25
setcolorsapt 8, 21
setpagedevice 16, 18, 20, 21
showpage 8