Acrobat Distiller Parameters

Technical Note #5151

Version : Acrobat 5.0

ADOBE SYSTEMS INCORPORATED

Corporate Headquarters

345 Park Avenue

San Jose, CA 95110-2704

(408) 536-6000

February 19, 2001

Contents

Preface

Adobe Acrobat Distiller converts PostScript language page descriptions into Portable Document Format (PDF) files, which are the native format of the Acrobat family of products. This technical note describes the parameters you can set to control the PDF file output.

Purpose

This technical note serves as a reference to the Distiller parameters.

Audience

The audience of this technical note includes developers who are developing products that work with Acrobat Distiller and end users who are using the Distiller user interface. Readers should be familiar with Portable Document Format (PDF).

Use this technical note if you are a developer writing applications that programmatically control Distiller. By placing Distiller-specific operators in the PostScript language input file, your application can control a number of parameters Distiller uses during PostScript-to-PDF conversion. For example, to change downsampling behavior, your application would insert PostScript code to change the values of downsampling parameters.

If you are an end user with special requirements for PDF file output, you can use this technical note as a guide to customizing the distiller parameters contained in a job options file.

Contents

This technical note is organized as follows:

- Chapter 1, “How To Set Distiller Parameters,” provides a brief summary of how to set Distiller parameters through user interface job options and programmatically in the PostScript input file. The chapter also lists caveats you should observe when setting parameters.
- Chapter 2, “Using Distiller Parameters,” provides information on how to use the Distiller parameters. It points out caveats you should be aware of and illustrates concepts you should understand about the various parameter groups. Major sections are:
  - 2.1, “Using the General Parameters,“describes the controls provided by these parameters.

  - 2.2, “Using the Compression Parameters,” describes JPEG, Flate, and automatic compression of images and page compression.
  - 2.3, “Using the Font Parameters,” defines Distiller’s control over font embedding and subsetting.
  - 2.4, “Using the Color Conversion Parameters,” generally describes the functions of these parameters and provides details on the **ColorConversionStrategy** parameter.
  - 2.5, “Using the Advanced Job Options Parameters,” introduces Distiller controls when creating embedded job tickets. The section places special emphasis on four parameters: **DetectBlends**, **ParseDSCCommentsForDocInfo**, **PreserveEPSInfo**, and **CreateJobTicket**.
- Chapter 3, “Parameter Reference,” lists and describes each Distiller parameter available through the user interface (and a few not available) along with the corresponding user interface name. Each top-level section in this chapter corresponds to the controls in a Job Options panel of the Distiller user interface. These sections are:
  - 3.1, “General Parameters,” which lists the general parameters.
  - 3.2, “Compression Parameters,” which lists the parameters that control compression of color, grayscale, and monochrome images as well as pages (line art and text). This section is organized into three subsections.
  - 3.3, “Font Parameters,” lists the font parameters for embedding and subsetting fonts.
  - 3.4, “Color Conversion Parameters,” lists the color conversion parameters including: the profiles controlling color conversions; selecting a color conversion strategy; and preserving halftone, transfer function, overprint, and UCR and BG information in the PDF file.
  - 3.5, “Advanced Job Options,” lists the advanced job options, including: encoding binary streams; creating job tickets; overriding parameters; controlling overprint mode; and preserving DSC, EPS, and OPI information in the PDF file.
- Chapter 4, “Changes Since Last Revision,” lists the changes to this document since the last revision.

Other Useful Documentation

The following documents are referenced in this technical note. All of these documents are available on the Adobe Solutions Network Web site. The URL for this site is:

http://partners.adobe.com/asn/developers

- *PDF Reference, second edition, version 1.3* provides a description of the PDF file format, as well as suggestions for producing efficient PDF files. It is intended for application developers who plan to produce PDF files directly.
- *Portable Document Format: Changes From Version 1.3 to 1.4* provides PDF 1.4 updates to the PDF file format.

- *PostScript Language Reference, third edition* is the definitive programmer's reference for the syntax and semantics of the PostScript language, the imaging model, and the effects of the graphics operators.
- *Portable Job Ticket Format, version 1.1* (Technical Note #5620) describes the structure and contents of job tickets. Distiller can produce job tickets that are embedded in PDF files.
- *Color Separation Conventions for PostScript Language Programs* (Technical Note #5044) provides information on overprint mode strategy. (See the description of the OPM parameter.)
- *Open Prepress Interface (OPI) 1.3 Specification* provides details on OPI version 1.3. (See the description of the PreserveOPIComments parameter.)
- *Open Prepress Interface (OPI) Specification - Version 2.0* (Technical Note #5660) provides details on OPI version 2.0. (See the description of the PreserveOPIComments parameter.)

1 How To Set Distiller Parameters

This chapter describes how you can set Distiller parameters.

Distiller parameters can be set by choosing a job options file at the time the Distiller *job* (PostScript input file) is submitted to Distiller, or by inserting PostScript code into the job itself.

1.1 Choosing A Job Options File

You can control the initial job options file through the Distiller user interface, or, programmatically, through the various Distiller automation methods. This section provides a brief overview of job option control through the user interface. For details on automation methods, see the *Distiller API Reference*.

At the user interface level, you can set Distiller parameters by selecting a file of predefined job options, or by creating your own set of job options. Distiller job options control compression, font embedding, color management, and many other PDF file properties.

For most Distiller applications, you can select from four predefined job options files that are provided with Distiller and shown in Table 1.1, “Predefined job options files.” You cannot modify these predefined job options sets.

TABLE 1.1 Predefined job options files

eBook	Settings are for PDF files intended to be read onscreen—especially using laptops and smaller devices. It is recommended that you use this option with Distiller version 5.0 and higher rather than using the Screen option.
Press	Settings are for PDF files to be produced as a printed product with high-quality final output.
Print	Settings are for PDF files intended to be printed, digitally copied, published on a CD-ROM, or sent to a client as a publishing proof.
Screen	Settings are for PDF files intended to be viewed onscreen. The resulting PDF file has a resolution of 72-dpi. This job options file is provided for compatibility with the pre-5.0 Distiller ScreenOptimized job options file.

Alternately, you can start with the contents of any job options file, make modifications using a standard text editor, and save the results as a new file. Refer to the Acrobat Guide in Distiller online Help for additional information on controlling job options through the user interface.

1.2 Inserting PostScript Code Into the Job

Distiller contains a PostScript 3 interpreter with two Distiller-specific operators, **setdistillerparams** and **currentdistillerparams**.

NOTE: This document uses the term parameter to refer to an individual entry (key-value pair) in a PostScript dictionary that is passed to **setdistillerparams** or returned by **currentdistillerparams**.

The **currentdistillerparams** operator returns a dictionary containing key-value pairs for all Distiller parameters. Each execution of this operator allocates and returns a new dictionary.

NOTE: Where the value of a key is another dictionary, **currentdistillerparams** returns the key-value pairs set in that dictionary.

The **setdistillerparams** operator attempts to set one or more Distiller parameters. To set a parameter, you assign a value to a dictionary key and pass the dictionary as an operand to **setdistillerparams**, for example:

```
<</CompressPages true>> setdistillerparams
```

Where the value of a key is another dictionary, provide the keys you want to set in that dictionary, for example:

```
<</AutoFilterGrayImages true /GrayACSImageDict<<
/QFactor 0.25 /HSamples [1 1 1 1] /VSamples [1 1 1 1]>>
>> setdistillerparams
```

If a key does not exist in the implementation of the Distiller application, Distiller ignores the key. If the value is of the correct type and is supported by the Distiller implementation, Distiller replaces the current value with the one specified. Values of the correct type that are not supported by the Distiller implementation do not generate an error. Instead, Distiller substitutes the nearest achievable value. A value of the wrong type results in a typecheck error, and all parameter values are left unchanged.

Parameter values you set in a PostScript input file using the **setdistillerparams** operator override those set through the user interface when the job began. The parameters remain in effect for the duration of the current **save** level. (See Section 3.7.3 in the *PostScript Language Reference, third edition* for a discussion of the **save** and **restore** operators.)

This behavior largely depends on the value of the **LockDistillerParams** parameter at the time **setdistillerparams** is called. If true, Distiller ignores any parameters defined in the incoming PostScript file. Otherwise the Distiller parameters, if any, defined in the PostScript file override those defined by the job options.

At the end of the current job, Distiller restores the values to those present before the job began.

To enable PostScript files containing the **currentdistillerparams** or **setdistillerparams** Distiller operators to be used on PostScript devices such as printers that do not

implement these operators, you must add the following definitions to the beginning of the file:

```
/currentdistillerparams where {pop}
{userdict /currentdistillerparams {1 dict} put} ifelse
/setdistillerparams where {pop}
{userdict /setdistillerparams {pop} put} ifelse
```

This PostScript code sequence uses the existing **currentdistillerparams** and **setdistillerparams**, if present. If not, it defines **currentdistillerparams** to return an empty one-element dictionary, and **setdistillerparams** to be a NULL operation.

PostScript language programs that use these operators must not assume that any particular key is present in the dictionary returned by **currentdistillerparams**, or that **setdistillerparams** has any particular side effects.

1.3 Caveats for Setting Distiller Parameters

You should observe the following caveats when setting Distiller parameters:

- Distiller examines the parameters shown in Table 1.2, “Parameters that should be included in a job options file,” at the start of a job. The correct way to ensure that these parameters are defined properly is to include settings for them in the job options file. Distiller checks the remaining parameters throughout the job as needed.

TABLE 1.2 Parameters that should be included in a job options file

AlwaysEmbed	**CompatibilityLevel**
NeverEmbed	**Binding**
EmbedAllFonts	**ParseDSCComments**
MaxSubsetPct	**ParseDSCCommentsForDocInfo**
SubsetFonts	**CreateJobTicket**

- Two Distiller parameters, **Optimize** and **DoThumbnails**, can only be set in a job options file.
- Regardless of the value of the **DetectBlends** parameter, Distiller disables idiom conversion of gradients to smooth shading when **CompatibilityLevel** is less than 1.3.

2 Using Distiller Parameters

This chapter provides information on how to use the Distiller parameters. It points out caveats you should be aware of and illustrates concepts you should understand about the various parameter groups displayed in the Distiller user interface.

2.1 Using the General Parameters

The general parameters allow you a number of general controls when creating a PDF file, including:

- Specifying the version of Acrobat for file compatibility. You can create PDF files that are compatible with Acrobat 3.0, Acrobat 4.0, or Acrobat 5.0.
- Specifying other file and device settings such as resolution and optimization.

NOTE: If you create files with Acrobat 5.0 compatibility, the resulting PDF files may not be compatible with earlier Acrobat versions.

- Manipulating pages: rotating, binding, and selecting page ranges to be distilled.

See Table 3.1, "General parameters," for a description of each general parameter and the corresponding name in the user interface.

2.1.1 General Parameters That Must Be Present at the Beginning of the Job Options File

If you specify values for either of the general parameters, **DoThumbnails** or **Optimize**, you must include the parameter settings at the beginning of the job options file.

2.1.2 General Parameters Example

The PostScript example below disables compression of text and graphics

```
<</CompressPages false >> setdistillerparams
```

2.2 Using the Compression Parameters

You can have Distiller compress (encode) text and line art in your PDF files, as well as compress and resample color, grayscale, and monochrome bitmap images. This section describes Distiller's parameters for compressing and resampling images. Depending on the settings you choose, compression and resampling can significantly reduce the size of a PDF file with little or no loss of detail and precision. Through Distiller parameters, you can choose JPEG, Flate, or automatic compression.

NOTE: The terms encode and compress are synonymous. Either may appear in a parameter or dictionary name.

See the following tables for descriptions of each of the compression parameters:

- Table 3.2, "Color image compression parameters"
- Table 3.3, "Grayscale image compression parameters"
- Table 3.4, "Monochrome image compression parameters"
- Table 3.5, "Page compression parameter"

2.2.1 JPEG Compression

The JPEG (Joint Photographic Experts Group) compression method is suitable for grayscale or color images, such as continuous-tone photographs that contain more detail than can be reproduced onscreen or in print. JPEG is *lossy*, which means that it removes image data and may reduce image quality, but it attempts to reduce file size with the minimum loss of information. Because JPEG eliminates data, it can achieve much smaller file sizes than Flate compression, which is a lossless compression method. See Section 2.2.2, "Flate Compression," for details.

Table 2.1, "Distiller job options for JPEG compression," shows how Distiller sets the JPEG job option values through Distiller parameters.

TABLE 2.1 Distiller job options for JPEG compression

Distiller Parameter	Value Set
AutoFilterColorImages	false
AutoFilterGrayImages	false
EncodeColorImages	true
EncodeGrayImages	true
ColorImageFilter	DCTEncode
GrayImageFilter	DCTEncode
ColorImageDict	Medium (See Table 2.4.)
GrayImageDict	Medium (See Table 2.4.)

2.2.2 Flate Compression

Flate (also called ZIP) is a compression method that works well on images with large areas of single colors or repeating patterns, such as screen shots and simple images created with paint programs, and for black-and-white images that contain repeating patterns. Acrobat provides 4-bit and 8-bit Flate compression options. If you use 4-bit Flate compression with 4-bit images, or 8-bit Flate with 4-bit or 8-bit images, the Flate method is *lossless*, which means it does not remove data to reduce file size and so does not affect an image's quality. If, however, you use 4-bit Flate compression with 8-bit data, you can affect the image quality since data is lost.

Table 2.2, "Distiller job options for Flate compression," shows how Distiller sets the Flate job option values through Distiller parameters.

TABLE 2.2 Distiller job options for Flate compression

Distiller Parameter	Value Set
AutoFilterColorImages	false
AutoFilterGrayImages	false
EncodeColorImages	true
EncodeGrayImages	true
EncodeMonoImages	true
ColorImageFilter	FlateEncode
GrayImageFilter	FlateEncode
MonoImageFilter	FlateEncode
ColorImageDict	Medium (See Table 2.4.)
GrayImageDict	Medium (See Table 2.4.)

NOTE: Adobe's implementation of the Flate filter is derived from the zlib package of Jean-Loup Gailly and Mark Adler, whose generous assistance we gratefully acknowledge.

2.2.3 Automatic Compression

Automatic compression is an option for color or grayscale bitmap images. Distiller determines the proper handling of documents containing photographic and screen shot images. If Distiller finds sharp color changes in 8-bits-per-component images, it uses Flate compression; if the image has smooth color changes, Distiller uses JPEG.

NOTE: Table 2.3, "Automatic Distiller job options," shows how Distiller sets the automatic job option values through Distiller parameters.

TABLE 2.3 Automatic Distiller job options

Distiller Parameter	Value Set
AutoFilterColorImages	true
AutoFilterGrayImages	true
EncodeColorImages	true
EncodeGrayImages	true
ColorACSImageDict	Medium (See Table 2.4.)
GrayACSImageDict	Medium (See Table 2.4.)

2.2.4 Setting Compression of Color And Grayscale Images

The DCTEncode filter compresses color or grayscale images using JPEG compression. The filter bases compression on settings contained in the DCTEncode parameter dictionary. For a description of the DCTEncode parameter dictionary, see "DCTEncode Filter" in Section 3.13.3 of the *PostScript Language Reference, third edition*.

Distiller Settings And Resulting Image Quality

Distiller sets keys in the DCTEncode parameter dictionary to achieve five levels of image quality. Specifically Distiller sets the keys listed below. (For a description of all keys in the DCTEncode parameter dictionary, see Section 3.13.3 in the *PostScript Language Reference, third edition*.)

- **HSamples**
- **VSamples**
- **QFactor**
- **Blend**
- **ColorTransform**

Table 2.4, "Image compression quality," shows the values Distiller sets to achieve Minimum, Low, Medium, High, and Maximum image quality.

NOTE: The **Blend** parameter is used internally in Distiller's image compression algorithm. It is recommended that you retain the default value (1).

TABLE 2.4 Image compression quality

Quality	HSamples	VSamples	QFactor	Blend	Color Transform
Minimum	[2 1 1 2]	[2 1 1 2]	1.5	1	1
Low	[2 1 1 2]	[2 1 1 2]	0.8	1	1
Medium	[2 1 1 2]	[2 1 1 2]	0.55	1	1
High	[2 1 1 2]	[2 1 1 2]	0.25	1	1
Maximum	[1 1 1 1]	[1 1 1 1]	0.1	1	1

Customizing Color and Grayscale Image Compression

Distiller provides "Distiller-specific" parameters for customizing compression values for color and grayscale images. Each of the Distiller parameters below is of type dictionary and contains the same set of keys as in the DCTEncode dictionary:

- **ColorACSImageDict**
- **ColorImageDict**
- **GrayACSImageDict**
- **GrayImageDict**

You can customize color or grayscale image compression by setting your own values for any of the DCTEncode dictionary keys in the dictionaries listed above. If, for example, you want to customize color images, you would set your own values for keys such as **HSamples**, **QFactor**, and so forth, in **ColorImageDict**. Distiller applies the compression you choose to all colors or grayscale bitmap images in a PDF file. See Section 2.2.8, "Controlling Downsampling and Encoding For Each Sampled Image," for details on controlling compression of individual images in a file.

2.2.5 Setting Compression of Monochrome (Black And White) Images

Distiller uses a different selection of compression filters for monochrome image compression.

NOTE: With the exceptions of the **AntiAliasMonoImages** and **MonoImageDepth** parameters, the monochrome image compression parameters also can be applied to stencil masks created by the **imagemask** operator. Parameter behavior is the same in both cases. For details on **imagemask**, see the *PostScript Language Reference, third edition*.

CCITTFaxEncode (CCITT Group 4) compression typically yields the best compression of monochrome images. Distiller sets one key (**K** key) in the

CCITTFaxEncode parameter dictionary to -1 for two-dimensional compression. See "CCITTFaxEncode Filter" in Section 3.13.3 in the *PostScript Language Reference, third edition*, for details on all of the keys in the CCITTFaxEncode parameter dictionary.

The Distiller-specific parameter **MonoImageDict** is of type dictionary and is based on the CCITTFaxEncode parameter dictionary (that is, it contains the same keys).You can choose to set any of the keys in the CCITTFaxEncode parameter dictionary for the value of the **MonoImageDict** parameter to customize monochrome image compression.

2.2.6 For Details on Encoding Filters

Encoding provides a way to compress images using any one of several compression filters. This document does not present the details of the filters; however, you can find that information in Section 3.13.3 of the *PostScript Language Reference, third edition*.

2.2.7 Downsampling and Subsampling Images

Distiller can reduce the number of dots per inch in an image by downsampling or subsampling. When Distiller downsamples or subsamples an image, it combines pixels in a sample area to make one larger pixel. Given the resolution setting of the output device in dots per inch, Distiller combines pixels as needed to reduce the image resolution to the specified setting. In general, monochrome images do not need to be downsampled.

Methods Distiller Uses To Reduce Image Resolution

Distiller's uses the following subsampling and downsampling methods:

- Subsampling. Distiller chooses a pixel in the center of the sample area and replaces the entire area with that pixel at the specified resolution. Subsampling significantly reduces the distilling time compared with downsampling but results in images that are less smooth and continuous.
- Average downsampling. Distiller averages the pixels in a sample area and replaces the entire area with the average pixel color at the specified resolution.
- Bicubic downsampling. Distiller uses a weighted average to determine pixel color and usually yields better results than the simple averaging method of downsampling. Bicubic is the slowest but most precise method, resulting in the smoothest tonal gradations.

These techniques can be very useful in reducing the size and drawing time for images that are primarily viewed on a screen instead of printed. For example, by downsampling or subsampling an image from a typical printer resolution of 300 dots per inch to a typical monitor resolution of 72 dots per inch, the amount of data needed to represent an image is decreased by a factor of 16, and the image can be drawn on the screen much more quickly.

Setting The Threshold Resolution For Downsampling Images

Distiller provides the following parameters for setting the threshold resolution for downsampling color, grayscale, and monochrome images, respectively:

- **ColorImageDownsampleThreshold**
- **GrayImageDownsampleThreshold**
- **MonoImageDownsampleThreshold**

The downsampling threshold is the ratio of the input image resolution (specified by the value of the respective **ColorImageResolution**, **GrayImageResolution**, or **MonoImageResolution** Distiller parameter) to the output resolution. Downsampling may be performed on images whose resolutions are higher than the threshold value.

Say, for example, **ColorImageResolution** is 72 and **ColorImageDownsampleThreshold** is set to 1.5 for a color input image. The input image would not be downsampled unless it has an output resolution greater than 108 dots per inch:

```
trunc((72 * 1.5) + .5) = 108 dpi
```

In PostScript, you would set the downsample threshold as shown here:

```
<</ColorImageDownsampleThreshold 1.5 >> setdistillerparams
```

Threshold values must be between 1.0 through 10.0, inclusive, with a default value of 1.5. If you set the threshold out of range, it reverts to a default of 2.0.

When To Downsample or Subsample Images

You should downsample or subsample bitmap images when they contain more data than the output device supports. If your images are sampled at a higher resolution than the device supports, the resulting excess data increases the time it takes the device to process the image. Image quality is not improved.

2.2.8 Controlling Downsampling and Encoding For Each Sampled Image

If you want, you can separately control the downsampling and encoding of each sampled image in a PostScript file. To do this, you must make adjustments to the Distiller parameters in the file just before, and appropriate to, each image.

NOTE: Distiller never samples images to a higher resolution, only to a lower one.

2.2.9 Controlling Bit Depth

Distiller can also change the bit depth of an image, whether in conjunction with downsampling or not. *Bit depth* is the number of bits used to represent each *color component* of each sample of an image. (Red, for example, is a color component in an RGB image.) Image sampling can either decrease the bit depth (for example, from 8 bits per sample to 4 bits per sample) to save space, or increase the depth to provide *anti-aliasing*.

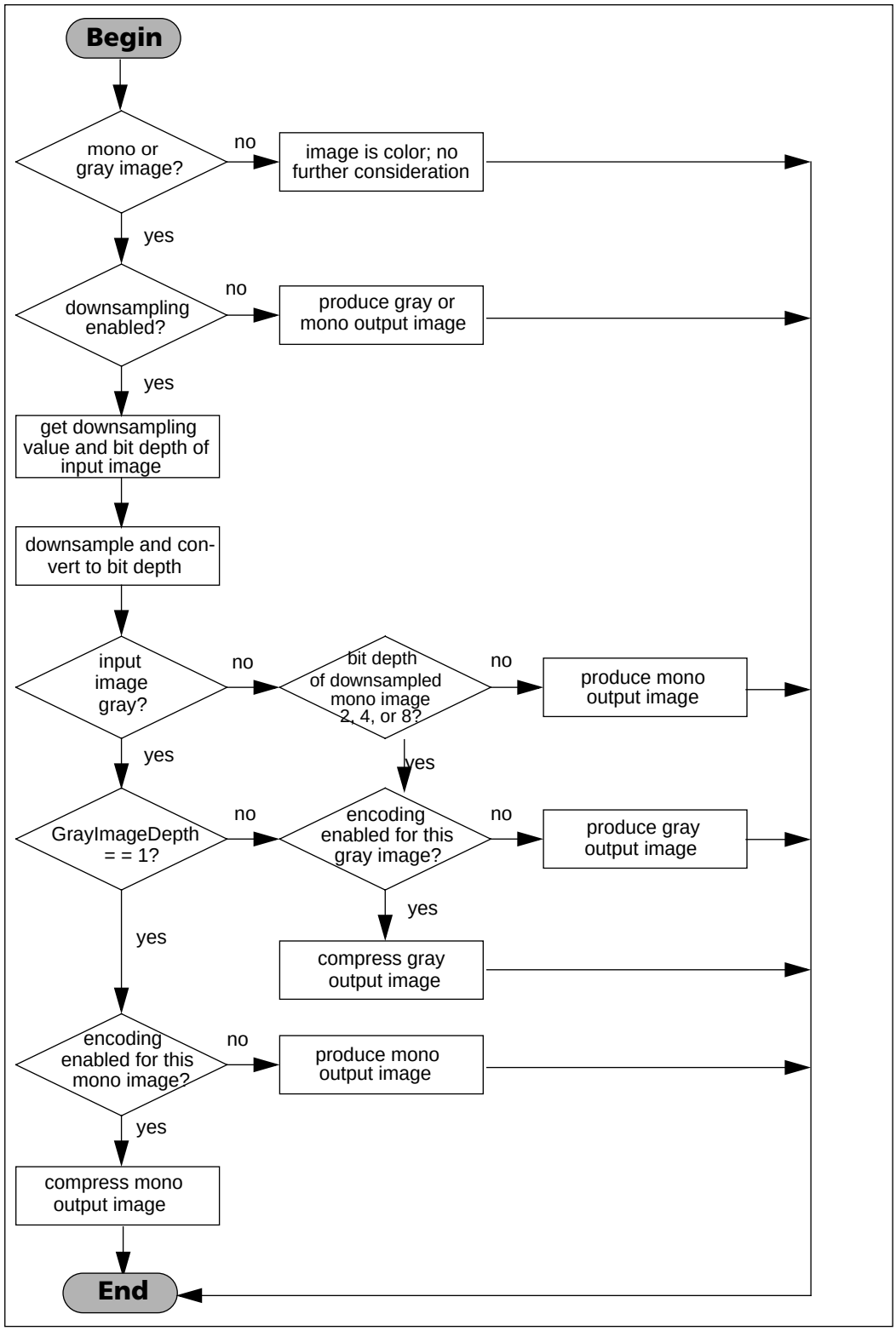
Anti-aliasing increases the number of bits per sample in downsampled images to preserve some of the information that is otherwise lost by downsampling. As an example of anti-aliasing, consider a 300-dpi monochrome image that is downsampled to 150 dpi. Such an image ends up as a 4-bit grayscale image, because each of the samples in the downsampled image is produced from four samples in the input image. Because each of the input samples can be either on or off, there are 16 possible values for each sample in the downsampled image. Although only three bits are needed to encode five values, the PDF file format only supports bit depths of 1, 2, 4, and 8 and hence the data are encoded using 4 bits per sample.

2.2.10 Caveats For Processing Grayscale and Monochrome Images

To correctly use the parameters for grayscale and monochrome sampled images, it is extremely important that you understand how Distiller processes these image types. It is easy to be confused because an input grayscale image specified to have a **GrayImageDepth** of 1 becomes a monochrome image, and a monochrome image that is downsampled and has a **MonoImageDepth** of 2, 4, or 8 becomes a grayscale image. In both cases, Distiller uses some parameters for monochrome images and some for grayscale images.

Distiller's image processing procedure is illustrated in Figure 2.1, "Processing grayscale and monochrome images."

FIGURE 2.1 Processing grayscale and monochrome images



The process shown in Figure 2.1 is described below:

1. Distiller determines whether the input image is monochrome, grayscale, or color. Unlike monochrome and grayscale images, a color image always remains a color image. Because the treatment of color images is unambiguous, Distiller does not consider them further.
2. Distiller checks the parameters to determine whether downsampling is enabled for the input image type and, if so, the value for the image downsampling resolution.
3. Distiller gets the image depth for the input image type by reading the Distiller parameter specifying it.
4. Distiller downsamples the image and converts it to the bit depth specified by the Distiller parameters, producing an output image. It is at this point in imaging processing that conversion between monochrome and grayscale images may occur, as described above.
5. Distiller checks the parameter specifying whether encoding is enabled for the output image type (not necessarily the same as the input image type).
6. If encoding is enabled, Distiller encodes the output image using the filter type and filter parameter dictionary specified by the Distiller parameters for the output image type

2.2.11 Compression Parameters Example

The PostScript example below shows a code fragment specifying that monochrome images be downsampled to 72-dpi, converted to 2 bits per sample, and encoded using Flate compression. Because the downsampled images are grayscale, the filter is specified using the grayscale rather than the monochrome image parameters. Also, assuming that the input image is a 300-dpi image, it is downsampled to 75-dpi, the closest possible value to the 72-dpi requested.

```
<<   /DownsampleMonoImages true
     /MonoImageResolution 72
     /MonoImageDepth 2
     /EncodeGrayImages true
     /AntiAliasMonoImages true
     /GrayImageFilter /FlateEncode
>> setdistillerparams
```

2.2.12 Setting Compression of Text and Line Art

You can use the **CompressPages** parameter to set the compression of text and line art.

2.3 Using the Font Parameters

Through Distiller parameters you can include fonts in PDF files (provided license restrictions are satisfied) to ensure that the Acrobat viewer can use the exact font for rendering the PDF file, regardless of whether the font is installed on the machine used to view the PDF file. You may, for example, need the exact font to achieve certain effects such as high-end printing or to ensure portability in situations where the viewer is unable to create a substitute font. Including a font in the PDF file is referred to as *embedding* the font in the file.

NOTE: You must check the license(s) for any font(s) you want to embed, to verify that embedding is allowed.

Embedded fonts make the PDF file larger. To produce files as small as possible, Distiller also has parameters enabling you to subset fonts. When you *subset* a font, Distiller includes only the information required to draw *glyphs* (or specific renderings, such as A or A for the abstract character A) for the characters used in the document. Subsetting is expressed as a percentage of the font glyphs for a font format.

Table 2.5 identifies the fonts that you can (or cannot) embed or subset through Distiller parameters.

TABLE 2.5 Distiller control over embedding and subsetting fonts

Font	NeverEmbed?	AlwaysEmbed?	Subset?
Type 1	Yes	Yes	Yes
Type 3	No - Always embedded		No - Always subsetted
True Type (Type 42)	Yes	Yes	No - Always subsetted
CIDFontType0	Yes	Yes	No - Always subsetted
CIDFontType1	No - Always embedded		No - Always subsetted
CIDFontType2	Yes	Yes	No - Always subsetted

See Table 3.6, “Font embedding and subsetting parameters,” for a description of each of the font parameters. For additional information on Type 1, Type 3, Type 42, and CID-keyed fonts, see Chapter 5, “Fonts,” in the *PostScript Language Reference, third edition*. You also can find additional documentation on fonts on the Adobe Solutions Network Web site:

http://partners.adobe.com/asn/developers

NOTE: Distiller 5 also supports OpenType fonts; Distiller 4 does not. OpenType fonts are based on the compact font format (CFF). For more information, see the *Compact Font Format Specification* on the Adobe Solutions Network Web site.

2.3.1 Font Embedding

Three parameters govern embedding entire fonts. Because these parameters can be confusing, they are discussed in this section.

- **AlwaysEmbed**
- **EmbedAllFonts**
- **NeverEmbed**

AlwaysEmbed and NeverEmbed

AlwaysEmbed and **NeverEmbed** are arrays of font names, the first array element of which may or may not be a boolean value. If a boolean value exists in the first array element and it is true, Distiller adds the fonts whose names are in the remaining array elements to its internal list of fonts to always embed (if this is the **AlwaysEmbed** array) or to its internal list of fonts to never embed (if this is the **NeverEmbed** array). The boolean value false causes Distiller to remove the fonts whose names are in the remaining array elements from its internal “always embed” or “never embed” font list, depending on whether this is the **AlwaysEmbed** or **NeverEmbed** array.

The absence of a boolean in the first array element (the array elements contain names only) sets the internal “always embed” or “never embed” font list to be exactly those fonts whose names are contained in the array elements.

EmbedAllFonts

EmbedAllFonts is a boolean that, when true, specifies that all fonts be embedded except those in the **NeverEmbed** array.

Parameter Priority

Given the discussion of the parameters above, you still might wonder what their priorities are when used. Distiller resolves ambiguity by giving the **NeverEmbed** list priority over the **EmbedAllFonts** flag and the **AlwaysEmbed** list. If, for example, a font is in the **NeverEmbed** list, it will not be embedded—even if the **EmbedAllFonts** flag is true or the font is in the **AlwaysEmbed** list.

2.3.2 Font Embedding Example

The PostScript example below specifies that Minion Regular always be embedded, and that ITC Stone Serif Italic and ITC Stone Sans never be embedded.

```
<< /AlwaysEmbed [ /Minion-Regular ]
     /NeverEmbed [ /StoneSans /StoneSerif-Italic ]
>> setdistillerparams
```

NOTE: The font name given to **definefont** does not have to match the name in the FontInfo dictionary. For instance, in this example the full name of the font defined as ‘StoneSans’ is ‘ITC Stone Sans.’

2.4 Using the Color Conversion Parameters

During color conversion, Distiller parameters allow you to control whether Distiller *preserves* (that is, passes into the PDF file) halftoning, overprinting, and transfer function information. Other color conversion parameters allow you to specify which ICC profile to use for the conversion. See “ICCBased Color Spaces” in the *PDF Reference, second edition,* for details on profiles.

2.4.1 ColorConversionStrategy Parameter Details

You set the **ColorConversionStrategy** parameter to select the ICC profile to use during the conversion process. This parameter requires a detailed explanation and is described here.

The **ColorConversionStrategy** parameter has four possible values:

- LeaveColorUnchanged. This is equivalent to selecting “Leave Color Unchanged” in the Distiller user interface.
- UseDeviceIndependentColor. This is equivalent to “Convert/Tag Everything for Color Management” in the Distiller user interface.
- UseDeviceIndependentColorForImages. This is equivalent to “Convert/Tag Only Images…” in the Distiller user interface.
- sRGB. This is equivalent to “Convert All Colors to CalRGB/sRGB” in the Distiller user interface.

NOTE: Separation and DeviceN color spaces in the input file are left unchanged in the PDF output.

Table 2.6 shows how Distiller converts the PostScript input to the equivalent color space for each of the **ColorConversionStrategy** parameter values.

In Table 2.6, *X/Y* entries mean “X in PDF 1.2 and Y in PDF 1.3 or higher.”

TABLE 2.6 PS color space (in) vs. PDF color space (out)

PS Input	LeaveColorUnchanged	UseDeviceIndependentColor	UseDeviceIndependentColorForImages	sRGB
Gray text and graphics	Gray	CalGray/ICCBased	Gray	Gray
Gray image	Gray	CalGray/ICCBased	CalGray/ICCBased	Gray
RGB text and graphics	RGB	CalGray/ICCBased	RGB	CalRGB/sRGB

TABLE 2.6 PS color space (in) vs. PDF color space (out)

PS Input	LeaveColorUnchanged	UseDeviceIndependentColor	UseDeviceIndependentColorForImages	sRGB
RGB image	RGB	CalGray/ICCBased	CalRGB/ICCBased	CalRGB/sRGB
CMYK text and graphics	CMYK	LAB/ICCBased	CMYK	CalRGB/sRGB
CMYK image	CMYK	LAB/ICCBased	LAB/ICCBased	CalRGB/sRGB
CIE text and graphics	Cal/ICC	Cal/ICC	Cal/ICC	CalRGB/sRGB
CIE image	Cal/ICC	Cal/ICC	Cal/ICC	CalRGB/sRGB

2.5 Using the Advanced Job Options Parameters

Advanced job options that you can customize include:

- Converting gradients to smooth shades
- Controlling whether Distiller creates an *embedded* job ticket (in the PDF file) that contains specific information such as trapping requirements to be passed to another application or to a printing device
- Creating job tickets

See Table 3.8, “Advanced job options parameters,” for a description of each of the advanced job option parameters. Four of the advanced job options discussed in this section are:

- **DetectBlends**
- **ParseDSCCommentsForDocInfo**
- **PreserveEPSInfo**
- **CreateJobTicket**.

2.5.1 Converting Gradients to Smooth Shades

DetectBlends deserves special mention. **DetectBlends** controls (enables or disables) the conversion of PostScript gradients to smooth shades.

Distiller uses two methods to perform the conversion of gradients to smooth shades:

- One method employs the PostScript LanguageLevel 3 feature called idiom recognition. Briefly, *idiom recognition* replaces certain procedures (or idioms) with others having equivalent behavior but producing better quality results. (See “Idiom Recognition” on page 119 of the *PostScript Language Reference, third edition*, for details.) The Distiller parameter **DetectBlends** enables the subset of idioms that detect gradients (or blends) for the following applications: Adobe Illustrator, Macromedia Freehand, Corel Draw, and Quark XPress.
- Distiller also converts gradients to smooth shades independently of idiom recognition. This method is application-independent, but it is less reliable than the first.

In Distiller 4.0, the blend detecting idioms (first method) are controlled by the **IdiomRecognition** user parameter, while the second method is controlled by **DetectBlends**. You must turn off **IdiomRecognition** to use **DetectBlends**.

In Distiller 5.0, **DetectBlends** also controls the blend detecting idioms. By default **IdiomRecognition** is turned on in Distiller 5.0; it is recommended that you leave it turned on and control the blend detecting idioms using the **DetectBlends** parameter.

2.5.2 ParseDSCCommentsForDocInfo Parameter Details

If **ParseDSCCommentsForDocInfo** is true, Distiller parses the document structuring convention (DSC) comments in the PostScript file and attempts to *preserve* (extract) the Document Information as properties of the PDF document. Table 2.7 lists this information.

TABLE 2.7 Document information

Document Information	Source
Author	from DSC keyword: **%%For:**
Creator	from DSC keyword: **%%Creator:**
Title	from DSC keyword: **%%Title:**
Producer	from Distiller product name ("Acrobat Distiller 5.0")
CreationDate	from Distiller time stamp (creation time of PDF file)
ModDate	from Distiller time stamp (creation time of PDF file)

Distiller 5.0 also embeds the Document Information as XML in the PDF file. To embed the information, Distiller adds a **Metadata** key in the Catalog dictionary whose value is an indirect reference to a metadata stream object. The metadata object contains the metadata (the Document Information) for the PDF document. The metadata is

represented as RDF, in conformance with the XAP Adobe Standard Metadata schema.

2.5.3 PreserveEPSInfo Parameter Details

If **PreserveEPSInfo** is true, Distiller attempts to preserve the encapsulated PostScript (EPS) information as properties of the PDF document. Table 2.8 lists this information.

TABLE 2.8 EPS information

Document Information	Source
Author	from DSC keyword: **%%For:**
Creator	from DSC keyword: **%%Creator:**
Title	from DSC keyword: **%%Title:**

Distiller 5.0 also embeds the information for embedded EPS files as XML in the PDF file. To do this, Distiller

- Adds a **Metadata** key in the property list of the marked content container for the EPS.
- Stores the property list as an indirect reference in the page resources object.

The value of the **Metadata** key is an indirect reference to the metadata stream object, which contains the metadata (the EPS information). The metadata is represented as RDF, in conformance with the XAP Adobe Standard Metadata schema.

2.5.4 CreateJobTicket Parameter Details

When the Distiller parameter **CreateJobTicket** is true, Distiller produces *internal job tickets* (that is, job tickets within the PDF file). Job ticket keys are created in response to **setpagedevice** keys and DSC comments.

The relationship between **setpagedevice** keys and job ticket keys, and the relationship between DSC comments and job ticket keys is described in the following sections. For details on the format and contents of job tickets, see *Portable Job Ticket Format, Version 1.1.*

Relationship Between setpagedevice Keys and Job Ticket Keys

Table 2.9, “Relationship between setpagedevice keys and job ticket keys,” lists the **setpagedevice** keys that Distiller supports and describes where in an internal job ticket Distiller stores the corresponding key values.

NOTE: **setpagedevice** keys that are distilled into the JobTicketContents dictionary rather than into the PageRange dictionary must appear in the first page of the

PostScript job; otherwise, they are ignored. In the **PS page** column of Table 2.9, "First" identifies **setpagedevice** keys that must appear on the first page.

TABLE 2.9 Relationship between setpagedevice keys and job ticket keys

setpagedevice key	PS page	job ticket key
Bind NOTE: **Bind** is unrelated to the Distiller parameter **Binding**.	First	**jobTicketContents::Finishing**
CutMedia	First	If the value of the **CutMedia setpagedevice** key is less than 4, Distiller represents the **setpagedevice** value in **JobTicketContents::MediaUsage::CutMedia.** Otherwise, it represents the value in **JobTicketContents::PrintLayout::Signature:: Sheets::MediaUsage::CutMedia.**
DeviceRenderingInfo/ ValuesPerColorComponent	Any	**PageRange::Rendering::ValuesPerColor Component**
Duplex	First	**JobTicketContents::PrintLayout** See Appendix B.4 in the *Portable Job Ticket Format, Version 1.1,* for a description of the general appearance of a job ticket that can produce duplex printing.
Fold	First	**JobTicketContents::Finishing**
HWResolution	Any	**PageRange::Rendering::Resolution**
Jog	First	**JobTicketContents::Finishing**
Laminate	First	**JobTicketContents::Finishing**
ManualFeed	First	**JobTicketContents::MediaSource::ManualFeed**
MediaClass	First	**JobTicketContents::MediaSource::MediaClass**
MediaColor	First	**JobTicketContents::MediaSource::MediaColor**
MediaPosition	First	**JobTicketContents::MediaSource::Position**
MediaType	First	**JobTicketContents::Media::Category**
MediaWeight	First	**JobTicketContents::Media::Weight**
MirrorPrint	First	**JobTicketContents::MediaUsage::MirrorPrint**
NegativePrint	First	**JobTicketContents::MediaUsage::NegativePrint**

TABLE 2.9 Relationship between setpagedevice keys and job ticket keys

setpagedevice key	PS page	job ticket key
PageSize	Any	**PageRange::MediaBox**
PostRenderingEnhance	Any	**PageRange::Rendering::PostRenderingEnhance**
PreRenderingEnhance	Any	**PageRange::Rendering::PreRenderingEnhance**
ProcessColorModel	Any	**PageRange::ColorModel::ProcessColorModel**
SeparationColorNames	Any	**PageRange::ColorModel::ColorantParams**
SeparationOrder	Any	**PageRange::ColorModel::ColorantOrder**
Separations	Any	**PageRange::ColorModel::Separations**
Staple	First	**JobTicketContents::Finishing**
Trapping	Any	**PageRange::Trapping::Trapping**
TrappingDetails	Any	**PageRange::Trapping::TrappingDetails**
Trim	First	**JobTicketContents::Finishing**
Tumble	First	**JobTicketContents::PrintLayout** Such a job ticket is identical to that described for the **Duplex setpagedevice** key, except the **CTM** for the **Back** surface is rotated 180 degrees.

Relationship Between PostScript Comments And Job Ticket Keys

When the Distiller parameter **ParseDSCComments** is true, Distiller interprets certain PostScript comments to produce true job ticket **PlaneOrder** objects. Such PostScript comments include **%%Page:** (which is more specifically a DSC comment), **%%QRKPageBegin:**, and **%%PlateColor:**. Distiller also supports the **%%PlateColor:** PostScript comment; however, use of that comment is discouraged.

3 Parameter Reference

This chapter describes each of the Distiller parameters. Parameters are grouped into the same categories you would find in the Distiller user interface.

3.1 General Parameters

Table 3.1 lists the Acrobat Distiller general parameters.The corresponding name, if any, in the General panel of the Distiller UI is shown just below the parameter name.

TABLE 3.1 General parameters

Parameter/ UI Name	**Type**	**Semantics**
AutoRotatePages Auto-Rotate Pages	name	Allows Distiller to try to orient pages based on the predominant text orientation. This parameter is only used if the file does not contain **%%ViewingOrientation**, **%%PageOrientation**, or **%%Orientation** DSC comments. If the file contains such DSC comments and the parameter **ParseDSCComments** is true, Distiller honors the comments. The order of precedence of the comments, from highest to lowest, is: **%%ViewingOrientation** **%%PageOrientation** **%%Orientation** The value of **AutoRotatePages** must be one of the following: *None* Distiller turns off **AutoRotatePages**. *All* Distiller takes the predominant text orientation across all pages and rotates all pages the same way. *PageByPage* Distiller rotates pages on a page-by-page basis. This value is useful for mixed portrait and landscape documents.

TABLE 3.1 General parameters

Parameter/ UI Name	Type	Semantics
Binding Binding	name	Controls the value of the **PageDirection** key in the ViewerPreferences dictionary of the PDF file. **PageDirection** determines how the printed pages would be bound. Values are: *Left* For Left Binding. *Right* For Right Binding.
CompatibilityLevel Compatibility	real	The PDF version number: 1.2, 1.3, or 1.4
CoreDistVersion	integer	*(Read only)* Version number of the Distiller implementation. This is neither the version number of the PostScript interpreter used in Distiller nor the version number displayed in the user interface. The *CoreDistVersion* number in the shipped version of Distiller is “5000” for Acrobat 5.0
DoThumbnails Generate Thumbnails	boolean	NOTE: This parameter must be present at the beginning of the job options file. If true, Distiller creates thumbnails for the pages of the resulting PDF file.
EndPage Page Range To	integer	**StartPage** and **EndPage** together determine the range of pages that are distilled. The default value for **StartPage** is 1, and the default value for **EndPage** is -1. If **StartPage** is greater than 1, no PDF output is produced for the first (**StartPage** -1) pages of PostScript. **StartPage** becomes page 1 of the PDF file. If **EndPage** is greater than -1, distilling stops after the **EndPage** of PostScript. Distiller checks these two parameters at the time that the first PostScript marking operator is executed in a job. NOTE: StartPage and EndPage are useful when debugging PostScript. They are not recommended for general purpose use, as Distiller does not retain page number references in document links.

TABLE 3.1 General parameters

Parameter/ UI Name	**Type**	**Semantics**
ImageMemory	integer	Number of bytes in the buffer used in the sample processing of color, grayscale, and monochrome images. When the buffer is full, Distiller writes its contents to disk.
OffOptimizations	integer	**NOTE:** This parameter does not appear in the user interface. It typically is used for debugging. Forces Distiller to turn off some optimizations internally. Setting individual bits in this integer achieves different results, as described below. *Bit 0* (0x00000001) Turns off duplicate image detection. Distiller tries to detect if two images (for example, background on every page) are the same. (This is time consuming and not always what the user may want.) *Bit 1* (0x00000002) Turns off image merging. Distiller tries to merge small neighboring images to a larger image. *Bit 2* (0x00000004) Turns off separating TrueType Roman characters into separate fonts with WinAnsi MacRoman encodings (in PDF 1.3 or later). *Bit 3 - 31* Reserved
Optimize Optimize PDF	boolean	**NOTE:** This parameter must be present at the beginning of the job options file. If true, Distiller optimizes the PDF file. See the *PDF Reference, second edition*, for more information on optimization (called *linearization* in that document).
StartPage Page Range From	integer	See the description of the **EndPage** parameter.

TABLE 3.1 General parameters

Parameter/ UI Name	**Type**	**Semantics**
UseFlateCompression	boolean	**NOTE:** This parameter is obsolete beginning with Distiller 4.0. This parameter, together with **CompatibilityLevel** and **CoreDistVersion**, determines the compression method Distiller uses for streams affected by **CompressPages**. If **CoreDistVersion** >= 4000, Distiller uses Flate compression. However, if **UseFlateCompression** is *false* and **CompressPages** is *true*, Distiller uses LZW compression regardless of the **CompatibilityLevel** setting.

3.2 Compression Parameters

Table 3.2, Table 3.3, Table 3.4, and Table 3.5 list the Acrobat Distiller compression parameters for the respective compression types:

- Color image
- Grayscale image
- Monochrome image
- Page (text and line art)

The corresponding name, if any, of each parameter in the Compression panel of the Distiller UI is shown just below the parameter name.

3.2.1 Color Image Compression Parameters

Table 3.2 lists the color image compression parameters. These parameters control the type of compression used and color conversion strategy as well as determine whether and how the image is downsampled.

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
AntiAliasColorImages	*boolean*	If true, Distiller permits anti-aliasing on color images. If false, anti-aliasing Distiller does not permit anti-aliasing. The default value is *false*. Anti-aliasing increases the number of bits per component in downsampled images to preserve some of the information that is otherwise lost by downsampling. Anti-aliasing is only performed if the image is actually downsampled and **ColorImageDepth** has a value greater than the number of bits per color component in the input image. For more information on anti-aliasing see Section 2.2.9, “Controlling Bit Depth.”
AutoFilterColorImages Compression	boolean	Distiller uses only if **EncodeColorImages** is true. If **AutoFilterColorImages** is true, Distiller uses JPEG for photographs and Flate for screen shots. If false, Distiller compresses all color sampled images using the filter specified by **ColorImageFilter**. The default setting for this option is true.
ColorACSImageDict Compression Quality	dictionary	Dictionary of parameters for JPEG compression when JPEG is chosen from the Automatic filter selection (see **AutoFilterColorImages**). **ColorACSImageDict** is based on the DCTEncode parameter dictionary described in *Section 3.13.3* in the *PostScript Language Reference, third edition*. See Section 2.2.4, “Setting Compression of Color And Grayscale Images,” for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
ColorImageDepth	integer	Specifies the number of bits per color component in the downsampled image when **DownsampleColorImages** is true. Allowed bit values are 1, 2, 4, and 8 (for 1, 2, 4, and 8 bits per color component) and –1 (which forces the downsampled image to have the same number of bits per color component as the original image).
ColorImageDict Compression Quality	dictionary	Dictionary of parameters for JPEG compression. **ColorImageDict** is based on the DCTEncode parameter dictionary described in *Section 3.13.3* in the *PostScript Language Reference, third edition*. See Section 2.2.4, "Setting Compression of Color And Grayscale Images," for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.
ColorImageDownsampleThreshold dpi for images above *value* dpi	number	Sets the downsample threshold for color images. This is the ratio of image resolution to output resolution above which downsampling may be performed. The default value is 1.5. Must be between 1.0 through 10.0, inclusive. If you set the threshold out of range, it reverts to a default of 2.0. See "Setting The Threshold Resolution For Downsampling Images," for details on using this parameter.
ColorImageDownsampleType Average Downsampling to Subsampling to Bicubic Downsampling to	name	Must be one of the following values: *Average* Distiller averages groups of samples to get the new downsampled value. *Bicubic* Distiller uses bicubic interpolation on a group of samples to get a new downsampled value. *Subsample* Distiller picks the center sample from a group of samples to get the new downsampled value.

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
ColorImageFilter Compression	name	Specifies the compression filter to be used for color images. Ignored if **AutoFilterColorImages** is true or **EncodeColorImages** is false. Valid names are: *DCTEncode* Selects JPEG compression. *FlateEncode* Selects Flate (ZIP) compression. If *DCTEncode* is specified, it is only used if the output image has 8 bits per color component, that is, • if **ColorImageDepth** is 8 • if **ColorImageDepth** is –1 and the original image has 8 bits per color component Otherwise **FlateEncode** is used.
ColorImageResolution dpi	integer	Specifies the resolution to which downsampled color images are reduced. A color image is downsampled if **DownsampleColorImages** is *true*, and the resolution of the input image meets the criteria described in "Setting The Threshold Resolution For Downsampling Images." Distiller downsamples to this minimum resolution. The legal values are from 9 to 2400, inclusive.
ConvertImagesToIndexed	boolean	If true, Distiller converts images that use fewer than 257 colors to an indexed colorspace for compactness. This conversion, when enabled, produces smaller PDF files but may make distillation slower.
DownsampleColorImages Compression	boolean	If true, Distiller downsamples color sampled images using the resolution specified by **ColorImageResolution**. If false, Distiller does not carry out downsampling, and the image resolution in the PDF file is the same as that in the source PostScript file.

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
EncodeColorImages Compression	boolean	If true, Distiller encodes color images using the compression filter specified by the value of the **ColorImageFilter** key. If false, Distiller does not apply any compression filters to color sampled images.

3.2.2 Grayscale Image Compression Parameters

Table 3.3 lists the grayscale image compression parameters. These parameters control the compression and downsampling of grayscale sampled images. The available parameters and their functions are identical to those for color sampled images.

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
AntiAliasGrayImages	boolean	If true, Distiller permits anti-aliasing on grayscale images. If false, Distiller does not permit anti-aliasing. The default value is false. Anti-aliasing increases the number of bits per sample in downsampled images to preserve some of the information that is otherwise lost by downsampling. Anti-aliasing is only performed if the image is actually downsampled and **GrayImageDepth** has a value greater than the number of bits per sample in the input image. For more information on anti-aliasing see Section 2.2.9, “Controlling Bit Depth.”
AutoFilterGrayImages Compression	boolean	Distiller uses only if **EncodeGrayImages** is true. If **AutoFilterGrayImages** is true, Distiller determines the compression algorithm for sampled grayscale images automatically, on an image-by-image basis; scanned photographs are compressed using JPEG, and screen shots are compressed using Flate. If false, Distiller compresses all grayscale sampled images using the filter specified by **GrayImageFilter**.

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
DownsampleGrayImages Compression	boolean	If true, Distiller downsamples grayscale images using the resolution specified by **GrayImageResolution**. If false, Distiller does not carry out downsampling, and the image resolution in the PDF file is the same as that in the source PostScript file.
EncodeGrayImages Compression	boolean	If true, Distiller encodes grayscale images using the compression filter specified by the value of the **GrayImageFilter** key. If false, Distiller does not apply any compression filters to grayscale sampled images.
GrayACSImageDict Compression Quality	dictionary	Dictionary of parameters for JPEG compression when JPEG is chosen from the Automatic filter selection (see **AutoFilterColorImages**). **GrayACSImageDict** is based on the DCTEncode parameter dictionary described in Section 3.13.3 in the *PostScript Language Reference, third edition*. See Section 2.2.4, "Setting Compression of Color And Grayscale Images," for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.
GrayImageDepth	integer	Specifies the number of bits per sample in the downsampled image when **DownsampleGrayImages** is true. Allowed values are 1, 2, 4, and 8 (for 1, 2, 4, and 8 bits per sample) and -1 (which forces the downsampled image to have the same number of bits per sample as the original image).
GrayImageDict Compression Quality	dictionary	Dictionary of parameters for JPEG compression. **GrayImageDict** is based on the DCTEncode parameter dictionary described in *Section 3.13.3* in the *PostScript Language Reference, third edition*. See Section 2.2.4, "Setting Compression of Color And Grayscale Images," for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
GrayImageDownsampleThreshold dpi for images above *value* dpi	number	Sets the image downsample threshold for gray scale images. This is the ratio of image resolution to output resolution above which downsampling may be performed. See "Setting The Threshold Resolution For Downsampling Images," for details on using this parameter.
GrayImageDownsampleType Average Downsampling to Subsampling to Bicubic Downsampling to	name	Must be one of the following values: *Average* Distiller averages groups of samples to get the new downsampled value. *Bicubic* Distiller uses bicubic interpolation on a group of samples to get a new downsampled value. *Subsample* Distiller picks the middle sample from a group of samples to get the new downsampled value.
GrayImageFilter Compression	name	Specifies the compression filter to be used for grayscale images. Distiller ignores if **AutoFilterGrayImages** is true or **EncodeGrayImages** is false. Valid names are: *DCTEncode* Selects JPEG compression. *FlateEncode* Selects Flate compression. If *DCTEncode* is specified, it is only used if the output image has 8 bits per sample, that is, • if the original image is a grayscale image and **GrayImageDepth** is 8 • if the original image is a grayscale image with 8 bits per sample and *GrayImageDepth* is –1 • if the original image is a monochrome image that is downsampled and **MonoImageDepth** is 8) Otherwise Distiller uses **FlateEncode**.

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
GrayImageResolution dpi	integer	Specifies the resolution to which downsampled gray images are reduced. A gray image is downsampled if **DownsampleGrayImages** is true, and the resolution of the input image meets the criteria described in "Setting The Threshold Resolution For Downsampling Images." Distiller downsamples to this minimum resolution. The legal values are from 9 to 2400, inclusive.

3.2.3 Monochrome Images Compression Parameters

Table 3.4 lists the monochrome image compression parameters. These parameters control the compression and downsampling of monochrome images. See *Section 2.2.5, "Setting Compression of Monochrome (Black And White) Images,"* for details on using these parameters.

NOTE: With the exception of the **AntiAliasMonoImages** and **MonoImageDepth** parameters, the compression parameters also can be applied to stencil masks created by the **imagemask** operator. Parameter behavior is the same in both cases. For details on **imagemask**, see the *PostScript Language Reference, third edition*.

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
AntiAliasMonoImages Anti-Alias to gray	boolean	If true, Distiller permits anti-aliasing on monochrome images. If false, Distiller does not permit anti-aliasing. The default value is *false*. Anti-aliasing increases the number of bits per sample in downsampled images to preserve some of the information that is otherwise lost by downsampling. Anti-aliasing is only performed if the image is actually downsampled and **MonoImageDepth** has a value greater than 1. For more information on anti-aliasing see Section 2.2.9, "Controlling Bit Depth." **NOTE:** Distiller does not do anti-aliasing for image masks, regardless of the value of **AntiAliasMonoImage**.

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	**Type**	**Semantics**
DownsampleMonoImages Compression	boolean	If true, Distiller downsamples monochrome images using the resolution specified by **MonoImageResolution**. If false, Distiller does not carry out downsampling, and the image resolution in the PDF file is the same as that in the source PostScript file.
EncodeMonoImages Compression	boolean	If true, Distiller encodes monochrome images using the compression filter specified by the value of the **MonoImageFilter** key. If false, Distiller does not apply any compression filters to monochrome images.
MonoImageDepth	integer	Specifies the number of bits per sample in the downsampled image when **DownsampleMonoImages** is true. Allowed values are 1, 2, 4, and 8 (for 1, 2, 4, and 8 bits per sample) and –1 (which forces the downsampled image to have the same number of bits per sample as the original image.) When **MonoImageDepth** is 2, 4, or 8, monochrome images are converted to grayscale images. Distiller does not use **MonoImageDepth** unless **AntiAliasMonoImages** is true. NOTE: Distiller ignores **MonoImageDepth** for image masks.
MonoImageDict Compression Quality	dictionary	Dictionary of parameters for CCITTFaxEncode compression. **MonoImageDict** is based on the CCITTFaxEncode parameter dictionary. See " Setting Compression of Monochrome (Black And White) Images," for details on Distiller's settings and customizing CCITTFaxEncode compression.
MonoImageDownsampleThreshold dpi for images above *value* dpi	number	Sets the image downsample threshold for monochrome images. This is the ratio of image resolution to output resolution above which downsampling may be performed. See "Setting The Threshold Resolution For Downsampling Images," for details on using this parameter.

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	Type	Semantics
MonoImageDownsampleType Average Downsampling to Subsampling to Bicubic Downsampling to	name	Must be one of the following values: *Average* Distiller averages groups of samples to get the new downsampled value. *Bicubic* Distiller uses bicubic interpolation on a group of samples to get a new downsampled value. *Subsample* Distiller picks the middle sample from a group of samples to get the new downsampled value.
MonoImageFilter Compression	name	Specifies the compression filter Distiller will use for monochrome images. Valid names are: *CCITTFaxEncode* Selects CCITT Group 3 or 4 facsimile encoding. *FlateEncode* Selects Flate compression. *R*unLengthEncode Selects run length encoding.
MonoImageResolution dpi	integer	Specifies the minimum resolution for downsampled monochrome images. This value is used only when **DownsampleMonoImages** is true and the resolution of the input image meets the criteria described in "Setting The Threshold Resolution For Downsampling Images." Distiller downsamples to this minimum resolution. The legal values are from 9 to 2400, inclusive.

3.2.4 Page Compression Parameters

Table 3.5 describes the page compression parameter. The UI Compression panel name for this option is Compress Text and Line Art.

TABLE 3.5 Page compression parameter

Parameters	Type	Semantics
CompressPages Compress Text and Line Art	boolean	If true, Distiller uses Flate compression to compress page content streams as well as form, pattern, and Type 3 font content streams.

3.3 Font Parameters

NOTE: You must check the license(s) for any font(s) you want to embed, to verify that embedding is allowed.

Table 3.6 lists the parameters available for controlling font embedding and subsetting. For more information on font embedding, see Section 2.3, "Using the Font Parameters."

TABLE 3.6 Font embedding and subsetting parameters

Parameter/ UI Name	**Type**	**Semantics**
AlwaysEmbed Always Embed	array	An array consisting either entirely of font names, or of a boolean followed by font names. Each font name must be the PostScript language name of the font (that is, the name given to **definefont**). Distiller treats the contents of **AlwaysEmbed** as follows: If the array consists entirely of names: • Distiller sets its internal list of fonts that must be embedded to be exactly the list of names in the array. If the first array value is a boolean: • If the value is true, Distiller adds the font names in the rest of the **AlwaysEmbed** array to its internal list of fonts that must be embedded. • If false, Distiller removes the font names in the rest of the **AlwaysEmbed** array from its internal list of fonts to be embedded. See **EmbedAllFonts** for an explanation of how the Distiller resolves ambiguity in the use of this parameter, **EmbedAllFonts**, and **NeverEmbed**.
CannotEmbedFontPolicy When Embedding Fails	name	The policy Distiller uses if it cannot find, or cannot embed, the font. The possible name values are: *OK* Distiller ignores and continues. *Warning* Distiller displays a warning and continues. *Error* Distiller quits distilling the current job.

TABLE 3.6 Font embedding and subsetting parameters

Parameter/ UI Name	**Type**	**Semantics**
EmbedAllFonts Embed All Fonts	boolean	If true, Distiller specifies that all fonts, except those in the **NeverEmbed** list, are to be embedded in the PDF file. **NOTE:** Distiller resolves ambiguity in the parameters that specify font embedding by giving the **NeverEmbed** list priority over the **EmbedAllFonts** flag and the **AlwaysEmbed** list.
MaxSubsetPct Subset embedded fonts when percent of characters used is less than *value* %	integer	The maximum percentage of glyphs in a font that can be used before the entire font is embedded instead of a subset. The allowable range is 1 through 100. Distiller only uses this value if **SubsetFonts** is true.
NeverEmbed Never Embed	array	An array consisting either entirely of font names, or of a boolean followed by font names. Each font name must be the PostScript language name of the font (that is, the name given to **definefont**). Distiller treats the contents of **NeverEmbed** as follows: If the array consists entirely of names: • Distiller sets its internal list of fonts that must never be embedded to be exactly the list of names in the array. If the first array value is a boolean: If the value is true, Distiller adds the font names in the rest of the **AlwaysEmbed** array to its internal list of fonts that must never be embedded. • If false, Distiller removes the font names in the rest of the **NeverEmbed** array from its internal list of fonts to never be embedded. See **EmbedAllFonts** in this section for an explanation of how the Distiller resolves ambiguity in the use of this parameter, **EmbedAllFonts**, and **NeverEmbed**. When creating the array of font names, you cannot add the font names to both the **AlwaysEmbed** and **NeverEmbed** lists. *See* **EmbedAllFonts** for an explanation of how the Distiller resolves ambiguity in the use of this parameter, **EmbedAllFonts**, and **NeverEmbed**.

TABLE 3.6 Font embedding and subsetting parameters

Parameter/ UI Name	**Type**	**Semantics**
SubsetFonts Subset embedded fonts	boolean	If true, Distiller enables font subsetting. If false, subsetting is not enabled. Font subsetting embeds only those glyphs that are used in a document, instead of the entire font. This reduces the size of a PDF file that contains embedded fonts. If font subsetting is enabled, Distiller determines whether to embed the entire font or a subset by the number of glyphs in the font that are used [including component glyphs referenced by 'seac' (Type 1) glyphs], and the value of **MaxSubsetPct**. Subsetted fonts in the PDF file appear with a 5-letter prefix and a plus (+) sign. For example, Palatino subsetted may appear as: `NPBOME+Palatino-Roman` NOTE: Embedded instances of multiple master fonts and of Type 3, TrueType, and CID fonts are always subsetted, regardless of the value of **SubsetFonts**.

3.4 Color Conversion Parameters

Table 3.7 lists the parameters color conversion parameters. See "ICCBased Color Spaces" in the *PDF Reference, second edition,* for details on profiles.

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	**Type**	**Semantics**
CalRGBProfile Working Spaces: RGB	string	The name of the ICC profile that is used for tagging or converting RGB images, text, and/or graphics.
CalCMYKProfile Working Spaces: CMYK	string	The name of the ICC profile that is used for tagging or converting CMYK images, text, and/or graphics.
CalGrayProfile Working Spaces: Gray	string	The name of the ICC profile that is used for tagging or converting Gray images, text, and/or graphics.

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	**Type**	**Semantics**
ColorConversionStrategy Color Management Policies	name	Sets the ICC profile used for color conversion strategy. See Section 2.4.1, "ColorConversionStrategy Parameter Details," for details on how to use this parameter.
DefaultRenderingIntent Intent	name	PostScript jobs may specify the rendering intent for objects by using the **findcolorrendering** and **setcolorrendering** operators, as described in Section 7.1.3 in the *PostScript Language Reference, second edition*. When the PostScript job does not specify rendering intent in this manner, Distiller determines the rendering intent written to the PDF by the value of **DefaultRenderingIntent**. If the value of this parameter is *Default* (the default value), no rendering intent is written to the PDF. Legal values of this parameter are: *Default* *Perceptual* *Saturation* *RelativeColorimetric* *AbsoluteColorimetric*
sRGBProfile	string	*(Read Only)* The name of the ICC profile that is used for converting device-dependent or device-independent color spaces to CalRGB (PDF 1.2) or sRGB (PDF 1.3 and above).
PreserveHalftoneInfo Preserve Halftone Information	boolean	If true, Distiller passes halftone screen information (frequency, angle, and spot function) into the PDF file. If false, halftone information is not passed in.
PreserveOverprintSettings Preserve Overprint Settings	boolean	If true, Distiller passes the value of the **setoverprint** operator through to the PDF file. Otherwise, overprint is ignored (Distiller does not pass the information into the PDF file.)

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	**Type**	**Semantics**
TransferFunctionInfo Transfer Functions	name	Determines how Distiller handles transfer functions, which are traditionally used to compensate for dot gain or dot loss that may occur when an image is transferred to film. For example, a file that is intended for output on a particular imagesetter may contain transfer functions that compensate for the dot gain inherent with that printer. Valid values are: *Preserve* Distiller preserves (passes into the PDF file) transfer functions. *Remove* Distiller ignores transfer functions. They are neither applied to the color values by Distiller nor passed into the PDF file. *Apply* Distiller uses the transfer function to modify the data it writes to the PDF file, instead of writing the transfer function itself to the file. This value is ignored by Distiller 4.0 but supported by Distiller 5.0. It is sometimes used to achieve artistic effects (although the *PostScript Language Reference* discourages such usage).
UCRandBGInfo Preserve Under Color Removal and Black Generation Settings	name	Tells Distiller whether to pass the arguments to **setundercolorremoval** and **setblackgeneration** into the PDF file. Must be one of the following values: *Preserve* Distiller preserves (passes into the PDF file) the arguments. *Remove* Distiller ignores the arguments. See Section 7.2.3 in the *PostScript Language Reference, third edition*, for details on the **setundercolorremoval** and **setblackgeneration** operators and descriptions undercolor removal (UCR) and black generation (BG).

3.5 Advanced Job Options

Table 3.8 list the advanced job option parameters that you can customize.

TABLE 3.8 Advanced job options parameters

Parameter/ UI Name	**Type**	**Semantics**
ASCII85EncodePages ASCII Format	boolean	If true, Distiller ASCII85 encodes binary streams such as page contents streams, sampled images, and embedded fonts, resulting in a PDF file that is pure ASCII. If false, Distiller does not encode the binary streams, resulting in a PDF file that may contain substantial amounts of binary data. Distiller checks the value of this parameter only once per document. Any change to it must be made before any marks are placed on the first page of the document.
AutoPositionEPSFiles Resize Page and Center Artwork for EPS Files	boolean	If true, Distiller resizes the created page to the size of the EPS file using the **%%BoundingBox** comment in the header of the file, and centers the EPS file on the page when the EPS file is distilled. Distiller ignores this parameter if **ParseDSCComments** is false.
CreateJobTicket Save Portable Job Ticket Inside PDF File	boolean	If true, Distiller creates a Job Ticket object in the PDF file that contains specific information about this file—such as trapping information—that can be passed along to another application or print device. This parameter pertains to Portable Job Ticket Format 1.1, as described in *Portable Job Ticket Format, version 1.1* (Technical Note #5620). See Section 2.5.4, “CreateJobTicket Parameter Details,” for details on this parameter.

TABLE 3.8 Advanced job options parameters

Parameter/ UI Name	**Type**	**Semantics**
DetectBlends Blends Detection	boolean	Converts gradients to smooth shades. If **DetectBlends** is *true* and **CompatibilityLevel** is 1.3 or higher, Distiller enables the conversion of gradients to smooth shades. If **DetectBlends** is *false*, Distiller disables conversion. Regardless of the **DetectBlends** value, if **CompatibilityLevel** is less than 1.3, Distiller disables conversion. Regardless of the **DetectBlends** and **CompatibilityLevel** values, Distiller disables gradient conversion to smooth shades if idiom recognition is turned off in the prologue file or in the PostScript file itself. See Section 2.5.1, "Converting Gradients to Smooth Shades," for a discussion of **DetectBlends** and idiom recognition.
EmitDSCWarnings Log DSC Warnings	boolean	If true, Distiller displays warning messages about questionable or incorrect DSC comments during the distillation of the PostScript file. Distiller ignores this parameter if **ParseDSCComments** is false.
LockDistillerParams Allow PostScript File to Override Job Options	boolean	If true, Distiller ignores any parameters defined in the incoming PostScript file (in this case, the Distiller parameters are defined by the job options as set in the user interface). If false, the Distiller parameters, if any, defined in the PostScript file override those defined by the job options set in the user interface.
OPM Illustrator Overprint Mode	integer	Controls the overprint mode strategy in the job. Set to 0 for full overprint or 1 for non-zero overprint. For more information, refer to Technical Note #5044, *Color Separation Conventions for PostScript Language Programs*.
ParseDSCComments Process DSC Comments	boolean	If true, Distiller parses the DSC comments for any information that might be helpful for distilling the file or for information that is passed into the PDF file. If false, Distiller treats the DSC comments as pure PostScript comments and ignores them.

TABLE 3.8 Advanced job options parameters

Parameter/ UI Name	**Type**	**Semantics**
ParseDSCCommentsForDocInfo Preserve Document Information from DSC	boolean	If true, Distiller parses the DSC comments in the PostScript file and attempts to preserve the Document Information as properties of the PDF document. In Distiller 4.0 and higher, Distiller places the Document Information in the Info dictionary of the PDF file; you can view the information in the File -> Document Info -> General window of the user interface. Distiller 5.0 also embeds the Document Information as XML in the PDF file. See Section 2.5.2, "ParseDSCCommentsForDocInfo Parameter Details," for details on this parameter. Distiller ignores this parameter if **ParseDSCComments** is false.
PreserveCopyPage Preserve Level 2 copypage Semantics	boolean	If true, Distiller maintains PostScript LanguageLevel 2 compatibility for the **copypage** operator. If false, Distiller uses the PostScript LanguageLevel 3 definition of the **copypage** operator. See the *PostScript Language Reference, third edition,* for more information.
PreserveEPSInfo Preserve EPS Information from DSC	boolean	If true, Distiller attempts to preserve the EPS information in the PostScript file as properties of the resulting PDF file. The distilled EPS content is identified as Marked Content using the **EmbeddedDocument** key. Distiller 5.0 also embeds the EPS information as XML in a Metadata dictionary attached to the Marked Content. See Section 2.5.3, "PreserveEPSInfo Parameter Details," for details on this parameter. Distiller ignores this parameter if **ParseDSCComments** is false.

TABLE 3.8 Advanced job options parameters

Parameter/ UI Name	**Type**	**Semantics**
PreserveOPIComments Preserve OPI Comments	boolean	If true, Distiller places the page contents within a set of Open Prepress Interface (OPI) comments in a Form XObject dictionary and preserves the OPI comment information in an OPI dictionary attached to the Form. Page contents data within a set of OPI comments may include proxy images, high-resolution images, or nothing. If **PreserveOPIComments** is false, Distiller ignores OPI comments and their page contents. Setting **PreserveOPIComments** to false results in slightly simpler and smaller PDF files. Doing so is acceptable when use of an OPI server is not anticipated. Distiller ignores **PreserveOPIComments** if **ParseDSCComments** is false. Distiller recognizes both OPI 1.3 and OPI 2.0. See the specifications for OPI 1.3 and 2.0 (TN #5660) on the Adobe Solutions Network Web site: http://partners.adobe.com/asn/developers
UsePrologue Use Prologue.ps and Epilogue.ps	boolean	If true, Distiller uses the *prologue.ps* file in the Data subdirectory and distills it prior to any PostScript job that is sent through. Distiller also distills the *epilogue.ps* file in the same directory after the same PostScript job is run. You can add any legal PostScript code and comments to these two files.

4 Changes Since Last Revision

This chapter describes the changes to the Acrobat Distiller Parameters document for Acrobat Version 5.

- Updated the description of **DetectBlends** parameter for Distiller Version 5.
- Reorganized the document contents. Created Chapter 2, “Using Distiller Parameters,” to separate the conceptual information from the Distiller parameter descriptions in Chapter 3, “Parameter Reference.”
- Added Figure 2.1, which illustrates in a flowchart the processing of grayscale and monochrome images.
- Expanded on image compression in Section 2.2.4, “Setting Compression of Color And Grayscale Images,” and in Section 2.2.5, “Setting Compression of Monochrome (Black And White) Images.”
- Updated the image compression settings in Table 2.4.
- Added the Acrobat 5.0 user interface names associated with the Distiller parameter names in Chapter 3, “Parameter Reference.”
- Updated the document for consistency and for technical accuracy.

4 Changes Since Last Revision

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

X

Z