



Acrobat Distiller Parameters

Adobe Developer Support

Technical Note #5151

05 December 1994

Adobe Systems Incorporated

Corporate Headquarters
1585 Charleston Road PO Box 7900
Mountain View, CA 94039-7900
(415) 961-4400 Main Number
(415) 961-4111 Developer Support
Fax: (415) 969-4138

Adobe Systems Europe B.V.
Europlaza
Hoogoorddreef 54a
1101 BE Amsterdam Z-O, Netherlands
+31-20-6511 355
Fax: +31-20-6511 313

Adobe Systems Eastern Region
24 New England
Executive Park
Burlington, MA 01803
(617) 273-2120
Fax: (617) 273-2336

Adobe Systems Japan
Swiss Bank House 7F
4-1-8 Toranomon, Minato-ku
Tokyo 105, Japan
+81-3-3437-8950
Fax: +81-3-3437-8968

Copyright © 1993, 1994 by Adobe Systems Incorporated. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the publisher. Any software referred to herein is furnished under license and may only be used or copied in accordance with the terms of such license.

PostScript is a trademark of Adobe Systems Incorporated. All instances of the name PostScript in the text are references to the PostScript language as defined by Adobe Systems Incorporated unless otherwise stated. The name PostScript also is used as a product trademark for Adobe Systems' implementation of the PostScript language interpreter.

Any references to a "PostScript printer," a "PostScript file," or a "PostScript driver" refer to printers, files, and driver programs (respectively) which are written in or support the PostScript language. The sentences in this book that use "PostScript language" as an adjective phrase are so constructed to reinforce that the name refers to the standard language definition as set forth by Adobe Systems Incorporated.

Adobe, the Adobe logo, Acrobat, Distiller, Adobe Type Manager, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions. Apple and Macintosh are registered trademarks and QuickDraw is a trademark of Apple Computer, Inc. Microsoft and MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation. UNIX is a trademark registered in the United States and other countries, licensed exclusively to X/Open Company, Ltd. X Window System is a trademark of the Massachusetts Institute of Technology. Helvetica and Times are trademarks of Linotype AG and/or its subsidiaries. ITC Zapf Dingbats and ITC Stone are registered trademarks of International Typeface Corporation. Other brand or product names are the trademarks or registered trademarks of their respective holders.

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 warranty 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.



Contents

Acrobat Distiller Parameters 5

- 1 Introduction 5
- 2 Reading and setting Distiller parameters 5
- 3 General parameters 6
- 4 Sampled images 7
 - Color images 8
 - Grayscale images 10
 - Monochrome images 11
 - Caveats for grayscale and monochrome images 12
- 5 Font embedding 13
- 6 Examples 15

Appendix: Changes Since Earlier Versions 17

Index 19

Acrobat Distiller Parameters

1 Introduction

The Adobe™ Acrobat™ Distiller™ application converts PostScript™ language page descriptions into Portable Document Format (PDF) files that are the native format of the Acrobat family of products. It is possible to control a number of parameters used in the conversion by placing Distiller-specific operators in the PostScript language input file. The parameters that can be set and read include those to control the compression of text and graphics, generation of thumbnail images, downsampling and encoding of sampled images, and embedding of Type 1 fonts and instances of Type 1 multiple master fonts. Image downsampling reduces the resolution of an image and is typically used to optimize images that will most frequently be seen on a monitor, where resolutions of 72-dpi are common, instead of printed, where resolutions of 300-dpi and greater are possible. Image encoding is done to compress images, reducing the amount of disk space needed to store them.

This note describes the Distiller parameters that can be controlled, and how to read and set them. In reading this note, it is helpful to have some understanding of the Acrobat family of products, as described in Technical Note #5152, *Adobe Acrobat Product Overview and Compatibility*, and some understanding of PDF, as described in the *Portable Document Format Reference Manual* (published by Addison–Wesley, ISBN 0-201-62628-4) and Technical Note #5156, *Updates to the Portable Document Format Reference Manual*.

2 Reading and setting Distiller parameters

The Acrobat Distiller application contains a Level 2 PostScript interpreter with two Distiller-specific operators, **setdistillerparams** and **currentdistillerparams**. The definitions of these operators are:

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

dict **setdistillerparams** –

Attempts to set one or more Distiller parameters using the key–value pairs contained in the dictionary passed as an operand. If a key does not exist in the implementation of the Distiller application, it is ignored. If the value specified for a key is of the correct type, but the specified value cannot be achieved by the Distiller implementation, the nearest achievable value is substituted without any error indication. If a value is of the wrong type, a **typecheck** error is generated and all parameters in the Distiller application are left unchanged. If the value is of the correct type, the current value is replaced with the specified value.

The values set using the **setdistillerparams** operator within a distilling job override those in effect when the job began, and remain in effect for the duration of the current **save** level (See Section 3.7.3 of the *PostScript Language Manual, Second Edition* for a discussion of **save** and **restore**). At the end of the current job, the values in are restored to those present before the job began.

In order to allow files containing these operators to be used on PostScript devices, such as printers, that do not implement the **currentdistillerparams** and **setdistillerparams** operators, the following definitions must be added to the beginning of each PostScript language program which uses either of these operators:

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

This sequence uses the existing **currentdistillerparams** and **setdistillerparams**, if present. If not, **currentdistillerparams** is defined so that it returns an empty one-element dictionary, and **setdistillerparams** is defined so that it is a null operation.

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

3 General parameters

Table 1 lists the general Distiller parameters. These can be used to determine the version of the Distiller application, enable or disable the generation of thumbnail images, and enable or disable the compression of text and graphics in a document.

Table 1 *General Distiller parameters*

<i>Parameter</i>	<i>Type</i>	<i>Semantics</i>
CoreDistVersion	Integer	(Read-only) Version number of the Distiller implementation. This is not the version number of the PostScript interpreter used in the Distiller application.
DoThumbnails	boolean	If <i>true</i> , a thumbnail image is generated for the page. If <i>false</i> , a thumbnail image is not generated. The value of this key is checked once per page of the document, allowing thumbnail generation to be selectively enabled and disabled for each page in a document, if desired. Any change to the value to be used for a page must be made before the first mark is made on that page.
LZWEncodePages	boolean	If <i>true</i> , the page contents streams, which contain the pages' text and graphics, are LZW encoded. If <i>false</i> , they are not. The value of this key is checked only once per document. Any change to it must be made before any marks are placed on the first page of the document. Encoding of sampled images is not controlled by LZWEncodePages , but by other parameters described in section 4.
ASCII85EncodePages	boolean	If <i>true</i> , page contents streams, sampled images, and embedded fonts are ASCII85 encoded, resulting in a PDF file that is almost pure ASCII. If <i>false</i> , they are not, resulting in a PDF file that may contain substantial amounts of binary data. The value of this key is checked only once per document. Any change to it must be made before any marks are placed on the first page of the document.

Note Turning off thumbnail generation can significantly increase the performance of the Distiller program.

4 Sampled images

The Distiller application provides a number of parameters to control downsampling and encoding of sampled images. Separate sets are provided for color, grayscale, and monochrome (black-and-white) images.

Downsampling reduces the number of dots per inch in the image and can be very useful in reducing the size and drawing time for images that will be primarily viewed on a screen instead of printed. For example, by downsampling an image from a typical printer resolution of 300-dpi to a typical monitor resolution of 72-dpi, the amount of data needed to represent an image is decreased by a factor of 16 (as explained in the discussion following Table 2) and the image can be drawn on the screen much more quickly.

When downsampling, the Distiller application can also change the bit depth of an image, either decreasing it (for example, from 8 bits per sample to 4 bits per sample) to save space, or increasing it in order to provide antialiasing. As an example of antialiasing, consider a 300-dpi monochrome image that is downsampled to 150-dpi. Such an image will end 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. Since each of the input samples can be either on or off, there are five possible levels for each sample in the downsampled image, corresponding to 0, 1, 2, 3, and 4 samples being “on” in the original 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.

Encoding provides a way to compress images using any one of several compression filters. The details of the filters are not presented here, but may be found in Section 3.13.3 of the *PostScript Language Reference Manual, Second Edition*, and in the *Portable Document Format Reference Manual*.

If desired, it is possible to individually control the downsampling and encoding of each sampled image in a file, by changing the Distiller parameters just before each image in the file.

4.1 Color images

Table 2 lists the Distiller parameters that are available to control the treatment of color sampled images. These parameters control the type of compression used and determine whether the image is downsampled.

Table 2 *Distiller parameters for color sampled images*

<i>Parameter</i>	<i>Type</i>	<i>Semantics</i>
DownsampleColorImages		
	boolean	If <i>true</i> , sampled color images are downsampled using the resolution specified by ColorImageResolution . If <i>false</i> , downsampling is not carried out, and the image resolution in the PDF file is the same as that in the source PostScript language file.
ColorImageResolution		
	integer	Specifies the minimum resolution for downsampled color images. This value is used only when DownsampleColorImages is <i>true</i> .
EncodeColorImages		
	boolean	If <i>true</i> , color images are encoded using the compression filter specified by the value of the ColorImageFilter key. If <i>false</i> , no compression filters are applied to color sampled images.

ColorImageFilter name Specifies the compression filter to be used for color images. Valid names are **DCTEncode** (to select JPEG compression) and **LZWEncode** (to select LZW compression).

ColorImageDict dictionary Dictionary of parameters for the JPEG encoding filter. See Section 3.13.3 of the *PostScript Language Reference Manual, Second Edition* and Technical Note #5116, *Supporting the DCT Filters in PostScript Level 2*, for details.

ColorImageDepth

integer Specifies the number of bits per color component in the downsampled image when **DownsampleColorImages** is *true*. Allowed 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.)

AntiAliasColorImages

boolean If *true*, antialiasing is permitted on color images. If *false*, antialiasing is not permitted. Antialiasing increases the number of bits per component in downsampled images to preserve some of the information that is otherwise lost by downsampling. Antialiasing will only be 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.

ConvertCMYKImagesToRGB

boolean If *true*, color images that are specified in the four-component CMYK color space are converted to the three-component RGB color space when they are placed in the PDF file. If *false*, CMYK images are not converted. Converting CMYK images to RGB reduces the size of PDF files containing such images. Because of the intermediate conversions used to transform CMYK images to RGB images, downsampled color images will have 8 bits per color component if **ColorImageDepth** is -1.

As mentioned in Table 2, the value of the **ColorImageResolution** key specifies the minimum resolution of the downsampled image. The actual resolution of the downsampled image must be an integer divisor of the original image resolution, and can be calculated from:

$$D = \text{int} \left(\frac{R_O}{R_D} \right)$$

$$R_{\text{Actual}} = \frac{R_O}{D}$$

where R_o is the resolution of the original image in the PostScript language file, R_D is the resolution specified by **ColorImageResolution**, R_{Actual} is the actual resolution of the downsampled image, and *int* is a function returning the integer part of its argument.

For example, if the original image has a resolution of 353 dpi and the value of **ColorImageResolution** is 150 dpi, the actual resolution of the downsampled image is:

$$D = \text{int}\left(\frac{353 \text{ dpi}}{150 \text{ dpi}}\right) = \text{int}(2.35) = 2$$

$$R_{Actual} = \frac{353 \text{ dpi}}{2} = 176.5 \text{ dpi}$$

4.2 Grayscale images

Table 3 lists the parameters available to 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 *Distiller parameters for grayscale sampled images*

Parameter	Type	Semantics
DownsampleGrayImages		
	boolean	If <i>true</i> , grayscale images are downsampled using the resolution specified by GrayImageResolution . If <i>false</i> , downsampling is not carried out, and the image resolution in the PDF file is the same as that in the source PostScript language file.
GrayImageResolution		
	integer	Specifies the minimum resolution for downsampled grayscale images. This value is used only when DownsampleGrayImages is <i>true</i> .
EncodeGrayImages		
	boolean	If <i>true</i> , grayscale images are encoded using the compression filter specified by the value of the GrayImageFilter key. If <i>false</i> , no compression filters are applied to grayscale sampled images.
GrayImageFilter	name	Specifies the compression filter to be used for grayscale images. Valid names are DCTEncode (to select JPEG compression) and LZWEncode (to select LZW compression).

GrayImageDict dictionary Dictionary of parameters for the JPEG encoding filter. See Section 3.13.3 of the *PostScript Language Reference Manual, Second Edition* and Technical Note #5116, *Supporting the DCT Filters in PostScript Level 2*, for details.

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.)

AntiAliasGrayImages

boolean If *true*, antialiasing is permitted on grayscale images. If *false*, antialiasing is not permitted. Antialiasing increases the number of bits per sample in downsampled images to preserve some of the information that is otherwise lost by downsampling. Antialiasing will only be performed if the image is actually downsampled and **GrayImageDepth** has a value greater than the number of bits per sample in the input image.

4.3 Monochrome images

Table 4 lists the parameters available to control the compression and downsampling of monochrome images. The parameters are substantially the same as those for color and grayscale sampled images, although a different selection of compression filters is available. In general, monochrome images do not need to be downsampled, and CCITT Group 4 compression typically yields the best compression, as described in Chapter 11 of the *Portable Document Format Reference Manual*.

Table 4 *Distiller parameters for monochrome sampled images*

Parameter	Type	Semantics
DownsampleMonolImages		
	boolean	If <i>true</i> , monochrome images are downsampled using the resolution specified by MonolImageResolution . If <i>false</i> , downsampling is not carried out, and the image resolution in the PDF file is the same as that in the source PostScript language file.
MonolImageResolution		
	integer	Specifies the minimum resolution for downsampled monochrome images. This value is used only when DownsampleMonolImages is <i>true</i> .
EncodeMonolImages		
	boolean	If <i>true</i> , monochrome images are encoded using the compression filter specified by the value of the MonolImageFilter key. If <i>false</i> , no compression filters are applied to monochrome images.

MonolImageFilter	name	Specifies the compression filter to be used for monochrome images. Valid names are LZWEncode (to select LZW compression), RunLengthEncode (to select run length encoding), and CCITTFaxEncode (to select CCITT Group 3 or 4 facsimile encoding).
MonolImageDict	dictionary	Dictionary of parameters for the CCITTFaxEncode encoding filter. See Section 3.13.3 of the <i>PostScript Language Reference Manual, Second Edition</i> for details.
MonolImageDepth	integer	Specifies the number of bits per sample in the downsampled image when DownsampleMonolImages is <i>true</i> . 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.) Note that when MonolImageDepth is 2, 4, or 8, monochrome images are converted to grayscale images.
AntiAliasMonolImages	boolean	If <i>true</i> , antialiasing is permitted on monochrome images. If <i>false</i> , antialiasing is not permitted. Antialiasing increases the number of bits per sample in downsampled images to preserve some of the information that is otherwise lost by downsampling. Antialiasing will only be performed if the image is actually downsampled and MonolImageDepth has a value greater than one.

4.4 Caveats for grayscale and monochrome images

In order to correctly use the parameters for grayscale and monochrome sampled images, it is extremely important to understand how the Distiller application processes images. Confusion can easily arise 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 **MonolImageDepth** of 2, 4, or 8 becomes a grayscale image. In both cases, some of the parameters used by the Distiller application are those for monochrome images and some are those for grayscale images, as described in the following paragraphs.

When the Distiller application encounters an image, the following series of steps occur:

1. A decision is made as to whether the input image is monochrome, gray-scale, or color. Unlike monochrome and grayscale images, a color image always remains a color image. Because the treatment of color images is unambiguous, they are not considered further here.
2. The Distiller parameters are checked to determine whether downsampling is enabled for the input image type and, if so, what the downsampled image resolution is.
3. The Distiller parameter specifying the image depth for the input image type is read.
4. The image is downsampled and converted to the bit depth specified by the Distiller parameters, producing an output image. This is the step in which a conversion between monochrome and grayscale images may occur, as described earlier in this section.
5. The Distiller parameter specifying whether encoding is enabled for the output image type (not necessarily the same as the input image type) is checked.
6. If encoding is enabled, the output image is encoded using the filter type and filter parameter dictionary specified by the Distiller parameters for the output image type.

5 Font embedding

The Distiller parameters allow control over the embedding of Type 1 fonts and instances of Type 1 multiple master fonts in PDF files. This ensures that the exact font can be used for rendering the PDF file, regardless of whether or not the font is installed on the machine used to view the PDF file.

Note You must check with the font vendor before embedding fonts, to ensure that it is permitted.

Table 5 lists the parameters available for controlling font embedding.

Table 5 Distiller parameters for Type 1 font embedding

Parameter	Type	Semantics
AlwaysEmbed		
	array of names	One or more names of fonts that are always to be embedded in the PDF file. Each name must be the PostScript language name of the font. Note that if the first element of the array is the boolean value <i>false</i> , the remaining array elements are names of fonts to be removed from, not added to, the AlwaysEmbed list.

NeverEmbed

array of names One or more names of fonts that are never to be embedded in the PDF file. Each name must be the PostScript language name of the font. Note that if the first element of the array is the boolean value *false*, the remaining array elements are names of fonts to be removed from, not added to, the **NeverEmbed** list.

EmbedAllFonts

boolean Specifies that all fonts, except those in the **NeverEmbed** list, are to be embedded in the PDF file.

SubsetFonts boolean If *true*, font subsetting is enabled. If *false*, it is not. Font subsetting embeds only those glyphs that are used, instead of the entire font. This reduces the size of a PDF file that contains embedded fonts. If font subsetting is enabled, the decision on whether to embed the entire font or a subset is determined by number of glyphs in the font that are used, and the value of **MaxSubsetPct**. Note that embedded instances of multiple master fonts are always subsetted, regardless of the setting of **SubsetFonts**.

MaxSubsetPct integer The maximum percentage of glyphs in a font that can be used before the entire font is embedded instead of a subset. This value is only used if **SubsetFonts** is *true*.

Note The **AlwaysEmbed** and **NeverEmbed** fonts lists are restored to their default values between each job.

Regardless of the setting of **EmbedAllFonts** and the font names present in the **AlwaysEmbed** list, the base 14 fonts listed in Table 6 are *never* embedded in PDF files.

Table 6 *Base 14 fonts*

Courier	Symbol
Courier-Bold	Times-Roman
Courier-Oblique	Times-Bold
Courier-BoldOblique	Times-Italic
Helvetica	Times-BoldItalic
Helvetica-Bold	ZapfDingbats
Helvetica-Oblique	
Helvetica-BoldOblique	

The Distiller application marks a Type 1 font or multiple master Type 1 font instance as a candidate for embedding in the resulting PDF file if:

- The font is a symbolic font (determined from various heuristics based on its encoding and glyph names), *or*
- **EmbedAllFonts** is true, *or*
- The font is in the **AlwaysEmbed** list

A font that is a candidate for embedding will be embedded in the PDF file unless:

- It is one of the base 14 fonts, *or*
- It is in the **NeverEmbed** list

As can be determined from these rules, if a font appears in both the **AlwaysEmbed** list and the **NeverEmbed** list, it will not be embedded. In addition, fonts in the **NeverEmbed** list are not embedded even if **EmbedAllFonts** is *true*.

*Note The decision whether or not a particular font will be embedded in the PDF file is made the first time the font is encountered in the PostScript language file. Subsequent occurrences of the font in the file do not cause the decision to be re-examined. Because of this, the addition of a font to either the **AlwaysEmbed** or **NeverEmbed** lists must be made before the font is first used in the file.*

6 Examples

Example 1 shows a code fragment that disables the generation of thumbnail images and disables LZW compression of text and graphics.

Note The examples in this section employ a Level 2 PostScript syntax using << and >> for constructing dictionaries.

Example 1 Disabling thumbnails and text & graphics compression

```
<< /DoThumbnails false
    /LZWEncodePages false
>> setdistillerparams
```

Example 2 shows a code fragment specifying that monochrome images are to be downsampled to 72-dpi, converted to 2 bits per sample, and encoded using the DCT (JPEG) filter. Note that, because the downsampled images are grayscale, the filter is specified using the grayscale, not the monochrome, image parameters. Also, assuming that the input image is a 300-dpi image, it will actually be downsampled to 75-dpi, the closest possible value to the 72-dpi requested, as described in section 4.1.

Example 2 *Antialiasing monochrome images*

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

Example 3 shows a code fragment specifying that Minion Regular should always be embedded, and that ITC Stone Serif Italic and ITC Stone Sans should never be embedded.

Example 3 *Type 1 font embedding*

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


Appendix: Changes Since Earlier Versions

Changes since June 19, 1993 version

Documented the additions made for version 2.0 of the Acrobat Distiller application:

- Added mention of Technical Note #5156, *Updates to the Portable Document Format Reference Manual*, to Section 1.
- Table 1— Added **ASCII85EncodePages**. Also changed discussion of **LZWEncodePages** to indicate that can no longer be set on a per-page basis, but must be appear before any marks are placed on the first page of the document.
- Table 2 — Added **ConvertCMYKImagesToRGB** and **AntiAliasColorImages**.
- Table 3 — Added **AntiAliasGrayImages**.
- Table 4 — Added **AntiAliasMonoImages**.
- Table 5 — Added **SubsetFonts** and **MaxSubsetPct**. Modified the description of **NeverEmbed** and **AlwaysEmbed** to describe how to remove fonts from these lists. In the note following the table, removed the discussion of using **save** and **restore** as an alternate (less direct) way to remove fonts from the **NeverEmbed** and **AlwaysEmbed** lists.
- Section 5 — Modified the discussion to include the embedding of multiple master Type 1 font instances. Added a note indicating that permission to embed fonts must be obtained from the font vendor.

Index

A

AlwaysEmbed 13
AntiAliasColorImages 9
AntiAliasGrayImages 11
antialiasing 8
AntiAliasMonolImages 12
ASCII85EncodePages 7

B

base 14 fonts 14

C

CCITTFaxEncode 12
ColorImageDepth 9
ColorImageDict 9
ColorImageFilter 9
ColorImageResolution 8
ConvertCMYKImagesToRGB 9
CoreDistVersion 7
currentdistillerparams 5

D

DCTEncode 9, 10
Distiller application 5
DoThumbnails 7
DownsampleColorImages 8
DownsampleGrayImages 10
DownsampleMonolImages 11
downsampling 7

E

EmbedAllFonts 14
EncodeColorImages 8
EncodeGrayImages 10
EncodeMonolImages 11

encoding images 8

F

fonts, base 14 14

G

GrayImageDepth 11
GrayImageDict 11
GrayImageFilter 10
GrayImageResolution 10
grayscale image, conversion to
monochrome 12

I

image resolution, calculating 9
images, encoding 8

L

LZWEncode 9, 10, 12
LZWEncodePages 7

M

MaxSubsetPct 14
monochrome image, conversion to
grayscale 12
MonolImageDepth 12
MonolImageDict 12
MonolImageFilter 12
MonolImageResolution 11

N

NeverEmbed 14

P

PDF 5

Portable Document Format 5

R

resolution, calculating 9

RunLengthEncode 12

S

setdistillerparams 6

SubsetFonts 14