

Adobe Acrobat 7.0



Acrobat Distiller Parameters

December 7, 2004



Adobe Solutions Network — <http://partners.adobe.com>

Copyright 2004 Adobe Systems Incorporated. All rights reserved.

NOTICE: All information contained herein is the property of Adobe Systems Incorporated. No part of this publication (whether in hardcopy or electronic form) may be reproduced or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of the Adobe Systems Incorporated.

PostScript is a registered 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.

Except as otherwise stated, any reference to a "PostScript printing device," "PostScript display device," or similar item refers to a printing device, display device or item (respectively) that contains PostScript technology created or licensed by Adobe Systems Incorporated and not to devices or items that purport to be merely compatible with the PostScript language.

Adobe, the Adobe logo, Acrobat, the Acrobat logo, Acrobat Capture, Distiller, PostScript, the PostScript logo and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple, Macintosh, and Power Macintosh are trademarks of Apple Computer, Inc., registered in the United States and other countries. PowerPC is a registered trademark of IBM Corporation in the United States. ActiveX, Microsoft, Windows, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. UNIX is a registered trademark of The Open Group. All other trademarks are the property of their respective owners.

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

Preface	7
Purpose	7
Audience	7
Contents	7
Other Useful Documentation	8
Conventions Used in This Document	10
Chapter 1 How to Set Distiller Parameters	11
Choosing an Adobe PDF Settings File	11
Inserting PostScript Code Into the Job	13
Caveats for Setting Distiller Parameters	15
Distiller Initialization	16
Distiller Processing Sequence	16
Using Distiller to Combine PostScript Files	16
How Distiller Handles Incorrect .joboptions Files	17
Chapter 2 Using Distiller Parameters	19
Using the General Parameters	19
Using the Image Compression Parameters	19
JPEG Compression	20
Flate Compression	20
JPEG2000 Compression	21
Automatic Compression	24
Setting Compression of Color and Grayscale Images	25
Setting Compression of Monochrome (Black and White) Images	27
For Details on Encoding Filters	28
Downsampling and Subsampling Images	28
Controlling Downsampling and Encoding for Each Sampled Image	32
Controlling Bit Depth	32
Caveats For Processing Grayscale and Monochrome Images	33
Overview of Distiller's Filter Selection Strategy for Images	35
Decision Flow for Color and Grayscale Image Compression	37

Use of Image Dictionaries for Color and Grayscale Image Compression.	38
Compression Parameters Example	38
Setting Compression of Text, Line Art, and Objects	38
Passing Through JPEG Images	39
Summary of Distiller Image Compression Controls.	40
Disabling of Image Cropping	41
Using the Font Parameters	41
Font Embedding	42
Font Embedding Example	43
Using the Color Conversion Parameters.	44
ColorConversionStrategy Parameter Details	44
Using the Advanced Adobe PDF Parameters.	45
Converting Gradients to Smooth Shades	46
ParseDSCCommentsForDocInfo Parameter Details.	46
PreserveEPSInfo Parameter Details	47
CreateJobTicket Parameter Details	47
CreateJDFFile Parameter Details	49
Using the Standards Parameters	50
Using the CheckCompliance, PDFX1aCheck, and PDFX3Check Parameters	50
Using the PDFXOutputIntentProfile Parameter	52
Chapter 3 Parameter Reference	55
General Parameters	56
Image Compression Parameters	60
Color Image Compression Parameters.	61
Grayscale Image Compression Parameters	69
Monochrome Images Compression Parameters.	78
Page Compression Parameter	84
Font Parameters.	84
Color Conversion Parameters	88
Advanced Adobe PDF Settings	93
Standards Parameters (Acrobat Professional Only)	100
Chapter 4 Changes Since Last Revision	105
Changes Made for Acrobat 5.	105
Changes Made for Acrobat 6.	105
Changes Made for Acrobat 7.	106

Appendix A Conversions Related to JDF	109
Basic JDF File Created	109
Representing Certain PostScript Keys as JDF Elements and Attributes	110
Converting Between the Linear Representation of <code>setpagedevice</code> Keys and Their Hierarchical JDF Counterparts	113
Mapping of Certain DSC Comments into JDF Elements and Attributes	113
Composite Jobs	114
Pre-Separated Jobs with Interleaved Separations	114
Pre-separated Single-colorant Jobs	114
Mapping of Parameters into JDF Elements and Attributes	114
General	115
Image Compression	116
Page Compression	118
Fonts	119
Color Conversion	119
Advanced	121
PDF/X	122
Conversion of Parameters Not Available Through UI	123
Index	125

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 (UI). 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 an Adobe PDF settings 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 the Adobe PDF settings file 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:
 - [Using the General Parameters](#) describes the controls provided by these parameters.

- [Using the Image Compression Parameters](#) describes JPEG, JPEG2000, Flate, and automatic compression of images and page compression.
- [Using the Font Parameters](#) defines Distiller's control over font embedding and subsetting.
- [Using the Color Conversion Parameters](#) generally describes the functions of these parameters and provides details on the **ColorConversionStrategy** parameter.
- [Using the Advanced Adobe PDF Parameters](#) introduces Distiller controls when creating embedded job tickets, and other advanced topics. The section places special emphasis on six parameters: **DetectBlends**, **ParseDSCCommentsForDocInfo**, **PreserveEPSInfo**, **CreateJobTicket**, **AllowTransparency**, and **CreateJDFFile**.
- [Using the Standards Parameters](#) provides details on producing PDF/A- and PDF/X-compliant files.
- [Chapter 3, “Parameter Reference,”](#) lists and describes each Distiller parameter available through the UI (and a few not available through the UI) along with the corresponding UI name. Each top-level section in this chapter corresponds to the controls in an Adobe PDF settings panel of the Distiller UI. These sections are:
 - [General Parameters](#) lists the general parameters.
 - [Image Compression Parameters](#) 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 four subsections.
 - [Font Parameters](#) lists the font parameters for embedding and subsetting fonts.
 - [Color Conversion Parameters](#) lists the color conversion parameters including: the profiles controlling color conversions; selecting a color conversion strategy; and preserving halftone, transfer function, and UCR and BG information in the PDF file.
 - [Advanced Adobe PDF Settings](#) lists the advanced Adobe PDF settings, including: encoding binary streams; creating job tickets; overriding parameters; controlling overprint mode; and preserving DSC, EPS, and OPI information in the PDF file.
 - [Standards Parameters \(Acrobat Professional Only\)](#) lists the parameters that are used to produce PDF/X-compliant PDF files.
- [Chapter 4, “Changes Since Last Revision,”](#) lists the changes to this document since previous revisions.
- [Appendix A](#) describes how the Adobe Normalizer product interprets and converts Distiller parameters. It supplements information provided in [Using the Advanced Adobe PDF Parameters](#) at the end of [Chapter 2, “Using Distiller Parameters.”](#)

Other Useful Documentation

The documents listed in this section are referenced in this technical note.

The following documents are available with the Acrobat SDK. If you did not install the SDK but would like to, please visit the [Adobe Solutions Network Web site](#).

- *PDF Reference, fifth edition, version 1.6* 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. This document is also available from <http://partners.adobe.com/links/acrobat>.

- *Acrobat Distiller API Reference* (Technical Note #5158) describes the API that you can use to automate the generation of PDF documents on both Windows and Macintosh platforms.
- *pdfmark Reference Manual* (Technical Note #5150) describes how you can use the pdfmark operator to represent PDF features in PostScript language files in order to cause these features to appear when the corresponding PDF is generated. This document is also available from <http://partners.adobe.com/links/acrobat>.
- *Acrobat and PDF Library API Overview* provides a comprehensive overview of the public API to the Adobe Reader, Acrobat Professional and Acrobat Standard products, as well as all of the PDF Library APIs.
- *Acrobat and PDF Library API Reference* provides a comprehensive reference describing the public API to the Adobe Reader, Acrobat Professional and Acrobat Standard products, as well as the PDF Library APIs.

The documents below are also available on the [Adobe Solutions Network Web site](#), but they are not part of the Acrobat SDK.

- *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.)
- *The Compact Font Format Specification* (Technical Note #5176) provides details of a font format that is suitable for compactly representing one or more Type 1 or CID-keyed fonts.

The following document is available by special arrangement.

- *Using Adobe Normalizer Server, Version 6.0.4*, details the workings of the Adobe Normalizer Server. It is intended for OEMs developing software that incorporates Normalizer.

Conventions Used in This Document

Font	Used for	Examples
monospaced	Paths and filenames	C:\templates\mytmpl.fm
	Code examples set off from plain text	These are variable declarations: AVMenu commandMenu, helpMenu;
monospaced bold	Code items within plain text	The GetExtensionID method ...
	Literal values in reference documents	The enumeration terminates if proc returns false .
monospaced italic	Pseudocode	ACCB1 void ACCB2 ExeProc(void) { do something }
	Placeholders in code examples	AFSSimple_Calculate(<i>cFunction</i> , <i>cFields</i>)
blue	Live links to Web pages	The Adobe Solutions Network URL is: http://partners.adobe.com/asn/
	Live links to sections within this document	See Using the SDK .
	Live links to Distiller parameters and keys within this document	Test whether an ASAtom exists.
bold	PostScript language and PDF operators, Distiller parameters, keywords, dictionary key names	The setpagedevice operator
italic	Document titles that are not live links	<i>Acrobat and PDF Library API Overview</i>
	New terms	<i>User space</i> specifies coordinates for...
	PostScript variables	<i>filename</i> deletefile

This chapter describes how you can set Distiller parameters and provides some specific user information for Distiller.

In general, Distiller parameters can be set by choosing an Adobe PDF settings file at the time the Distiller *job* (PostScript input file) is submitted to Distiller, or by inserting PostScript code into the job itself.

Choosing an Adobe PDF Settings File

You can control the Adobe PDF settings file through the Distiller user interface (UI), or you can control Distiller programmatically through the various Distiller automation methods. This section provides a brief overview of Adobe PDF settings control through the UI. For details on automation methods, see the *Acrobat Distiller API Reference*, available from [Adobe Solutions Network Web site](#).

At the user interface level, you can set Distiller parameters by selecting a file of predefined Adobe PDF settings, or by creating your own set of Adobe PDF settings. Each Distiller settings file defines compression, font embedding, color management, PDF/X-compliance, and many other properties for all PDF files that Distiller creates. Distiller comes with several predefined Adobe PDF settings files, with settings defined for common scenarios. These are shown in Table 1.1, “[Predefined Adobe PDF settings files](#).”

NOTE: For historical reasons, the compatibility level in these files is set to PDF 1.4 rather than PDF 1.5. New versions of Distiller are historically set by default to generate PDF files that are backward compatible with an earlier version of the product.

TABLE 1.1 *Predefined Adobe PDF settings files*

Standard	These settings are the default. Use these settings to create PDF documents suitable for reliable viewing and printing of business documents. The PDF documents can be opened with Acrobat and Adobe Reader 5.0 and later.
High Quality Printing	Use these settings to create PDF documents for quality printing on desktop printers and proofers. The PDF documents can be opened with Acrobat and Adobe Reader 5.0 and later.
Press Quality	Use these settings to create PDF documents with improved image resolution for high quality pre-press printing. The PDF documents can be opened with Acrobat and Adobe Reader 5.0 and later. These settings require font embedding.

TABLE 1.1 *Predefined Adobe PDF settings files*

Smallest File Size	Use these settings to create PDF documents with minimum image resolution and no font embedding. The PDF documents can be opened with Acrobat and Adobe Reader 5.0 and later.
PDFA Draft	Use these settings to create PDF documents that are to be checked or must conform to PDF/A, a proposed ISO standard for the long-term preservation (archival) of electronic documents. For more information on creating PDF/A compliant PDF documents, please refer to the <i>Acrobat Guide</i> in Distiller online Help. Created PDF documents can be opened with Acrobat or Adobe Reader 5.0 and later.
PDFX1a 2001	(<i>In Acrobat 6, this was the PDFX1a PDF settings file</i>) Use these settings to create PDF documents that are to be checked or must conform to PDF/X-1a:2001, an ISO standard for graphic content exchange. For more information on creating PDF/X-1a compliant PDF documents, please refer to the <i>Acrobat Guide</i> in Distiller online Help. Created PDF documents can be opened with Acrobat or Adobe Reader 4.0 and later.
PDFX1a 2003	Use these settings to create PDF documents that are to be checked or must conform to PDF/X-1a:2003, an ISO standard for graphic content exchange. For more information on creating PDF/X-1a compliant PDF documents, please refer to the <i>Acrobat Guide</i> in Distiller online Help. Created PDF documents can be opened with Acrobat or Adobe Reader 5.0 and later.
PDFX3 2002	(<i>In Acrobat 6, this was the PDFX3 PDF settings file</i>) Use these settings to create PDF documents that are to be checked or must conform to PDF/X-3:2002, an ISO standard for graphic content exchange. For more information on creating PDF/X-3 compliant PDF documents, please refer to the <i>Acrobat Guide</i> in Distiller online Help. Created PDF documents can be opened with Acrobat or Adobe Reader 4.0 and later.
PDFX3 2003	Use these settings to create PDF documents that are to be checked or must conform to PDF/X-3:2003, an ISO standard for graphic content exchange. For more information on creating PDF/X-3 compliant PDF documents, please refer to the <i>Acrobat Guide</i> in Distiller online Help. Created PDF documents can be opened with Acrobat or Adobe Reader 5.0 and later.

NOTE: Before creating an Adobe PDF file to send to a commercial printer or a service bureau, check with the provider to find out what the output resolution and other settings should be, or ask him to provide a `.joboptions` file containing his recommended settings. You may need to customize the settings for a particular provider and then provide him with a `.joboptions` file of your own.

You should not modify any of the predefined Adobe PDF settings files. But you can start with these and create new ones based on them, or define and save your own sets of options and use them at any time. You can build your own Adobe PDF settings files using a standard text editor and save them in a folder of your choice, or you can open an existing settings file from the UI and modify it using the UI. However, if you choose to modify a settings file from the UI itself (as opposed to editing the file with a text editor), you will not be able to set every Distiller parameter to every possible value—the UI is designed to allow you to set parameters within a settings file to most possible values, but not all. In addition, some parameters cannot be set from the UI itself but require use of a standard text editor.

NOTE: In this document, when a “default” parameter value is provided, the value generally refers to what the value is “hard-coded” to within Distiller if the Adobe PDF settings file is empty or if that parameter or key is missing from the settings file. In general, the default, “hard-coded” settings are the same as those in the `Standard.joboptions` file, with a few exceptions (see “[Parameter Reference](#)” on page 55” for a complete listing of default values, and the exceptions).

Refer to the *Acrobat Guide* in Distiller online Help for additional information on controlling Adobe PDF settings through the UI.

Inserting PostScript Code Into the Job

Distiller contains a PostScript 3 interpreter with two Distiller-specific operators, **setdistillerparams** and **currentdistillerparams**. An Adobe PDF settings file can be thought of as a Postscript file that contains **setdistillerparams** key-value pairs followed by the PostScript **setpagedevice** operator with two key-value pairs. The Adobe PDF settings file is the first file processed by Distiller, but parameters can subsequently be set or changed within the PostScript job itself (see “[Distiller Processing Sequence](#)” on page 16).

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 /GrayACSIImageDict <<
/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. However, if Distiller writes out a new Adobe PDF settings file, it will still “pass through” the unknown key and value from the original Adobe PDF settings file to the new Adobe PDF settings file. If the value is of the correct type and is supported by the Distiller implementation, Distiller replaces the current value with the one specified. For values of the correct type that are not supported by the Distiller implementation, Distiller either generates an error or substitutes the nearest achievable value. A value of the wrong type results in a typecheck error, and all parameter values are left unchanged.

NOTE: Where appropriate, this document identifies the cases when Distiller emits a specific error message or when it handles range errors in a certain way.

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.)

(Note, however, that this behavior 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 Adobe PDF settings file.)

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.

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 an Adobe PDF settings 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 Adobe PDF settings file. Distiller checks the remaining parameters throughout the job as needed.

TABLE 1.2 *Parameters that should be included in an Adobe PDF settings file*

AlwaysEmbed	CompatibilityLevel
NeverEmbed	Binding
EmbedAllFonts	ParseDSCComments
MaxSubsetPct	ParseDSCCommentsForDocInfo
SubsetFonts	CreateJobTicket
AutoRotatePages	ColorConversionStrategy
LockDistillerParams	Description
PDFX1aCheck	PDFX3Check
PDFXCompliantPDFOnly	PDFXNoTrimBoxError
PDFXTrimBoxToMediaBoxOffset	PDFXSetBleedBoxToMediaBox
PDFXBleedBoxToTrimBoxOffset	PDFXOutputIntentProfile
PDFXTrapped	CheckCompliance
PreserveFlatness (see note below)	—

- **PreserveFlatness** cannot be changed in the middle of a page using **setdistillerparams**.

This parameter must be set before any marks are created on the page. For example, when the following PostScript command sequence:

```
1 1 moveto 1 2 lineto
<< /PreserveFlatness false >> setdistillerparams
20 setflat
```

is distilled, the value in the `.joboptions` file will be honored and the changes indicated by **setdistillerparams** will not be used. Any attempt to change **PreserveFlatness** after any marks are drawn on a page is ignored.

- If you specify values for the parameters **DoThumbnails**, **Optimize**, and/or **CompressObjects**, you must include the parameter settings in the Adobe PDF settings file and not in the PostScript input file itself. The parameters are not used until the post-processing step of distillation.

- Regardless of the value of the **DetectBlends** parameter, Distiller disables idiom conversion of gradients to smooth shading when **CompatibilityLevel** is less than 1.3.

Distiller Initialization

Distiller initialization is controlled by the file `distinit.ps`, which is executed once at Distiller startup. As part of the initialization process, all of the files in the Distiller Startup folder are executed. `Example.ps` is one such file.

You can add as many other startup files as you want inside the Startup folder. Those files are executed by `distinit.ps` during initialization.

NOTE: The files in the Startup folder are not executed in any specific order—not alphabetically, nor in any other predictable order. For that reason, it is best to add all extra initialization code to the `Example.ps` file to ensure that it all runs in order.

Distiller Processing Sequence

The Adobe PDF settings file is the first file that Distiller processes in the Distiller processing sequence. The Distiller processing sequence is as follows:

1. Adobe PDF settings file
2. Prologue file (if the **UsePrologue** parameter is `true`)
3. PostScript file
4. Epilogue file (if the **UsePrologue** parameter is `true`)

If the **LockDistillerParams** parameter is set to `false`, then the **setdistillerparams** operator can be used within the Prologue, PostScript, and/or Epilogue files to assign Distiller parameters new values (but be sure to observe the caveats listed in “[Caveats for Setting Distiller Parameters](#)” on page 15”).

Using Distiller to Combine PostScript Files

NOTE: Adobe recommends that the Acrobat 7.0 feature Create PDF from Multiple Files be used to combine PostScript and/or PDF files together into one PDF.

Distiller can combine two or more PostScript files to produce a single Adobe PDF document. If the PostScript files have embedded font subsets, Distiller gives the resulting PDF file only one subset for each font. This is much more efficient than creating a set of several PDF documents with duplicate font subsets.

To use files provided as part of the Acrobat installation to combine two or more PostScript files to create a single PDF, follow these steps:

1. Start a text editor or a word processor.
2. Choose File > Open.
3. Use your operating system's search tool to locate and open one of the following files in a text editor. For Acrobat 7 on Windows, these files should be located in \Documents and Settings\All Users\Documents\Adobe PDF 7.0\Example Files (Windows 2000) or \Documents and Settings\All Users\Shared Documents\Adobe PDF 7.0\Example Files (Windows XP); on Mac, they should be at /Library/Application Support/Adobe PDF/Example Files.
 - Runfilex.ps combines a set of PostScript files from one or more locations into one PDF file.
 - Rundirex.txt combines a set of PostScript files from a specific folder or directory into one PDF file.
4. Follow the instructions in the sample files. The PostScript files are combined in the order in which they are listed.
5. Choose File > Save As, and save the modified files under a new name. Use the name you want Distiller to give the PDF file. For example, if you name the file Handbook.ps, Distiller creates a PDF file called Handbook.pdf. If you're using a word processor, save the file as a text file.
6. Quit the text editor or word processor.
7. Open the file in Distiller.
8. Convert the combined file to PDF. You can also place the file in an In folder to be converted later.
9. When the PDF file is ready, open the file in Acrobat and make sure that all of the document elements are present and in the correct order.

NOTE: The conversion settings used are those listed in the Default Settings pop-up menu in the Acrobat Distiller dialog box.

How Distiller Handles Incorrect .joboptions Files

Distiller uses several criteria when determining whether a .joboptions file is "correct." Distiller recognizes two cases of incorrectness:

1. The file is syntactically invalid (i.e., invalid PostScript).
2. The file is semantically invalid (i.e., invalid type or value).

Based on which case of incorrectness Distiller determines the .joboptions file to be, it takes one of several actions. In the first case, Distiller does not allow PDF creation. This applies to both the GUI application and the Watched Folder implementation. (For more

information on Watched Folders, please refer to the *Acrobat Guide* in Distiller online Help.) In the second case, Distiller takes one of three parameter-specific actions:

1. Distiller does not allow PDF creation.
2. Distiller uses a default value and ignores the semantic error.
3. Distiller resets all values to defaults and ignores the semantic error.

The first action is intended for problems for which a workaround is not reasonable, given that the resulting PDF file would contain unexpected content.

The second action is intended for parameter values whose type is correct but whose incorrect value can be reasonably or predictably converted to another value that does not affect the intent. For example: **/GrayImageFilter /LZWEncode** is taken to be **/GrayImageFilter /FlateEncode**.

The third action is intended for more seriously incorrect parameters, but not so severe as to require that the PDF file not be created. Many incorrect parameters cause Distiller to take this course of action.

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 UI.

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 (PDF 1.2), 4.0 (PDF 1.3), 5.0 (PDF 1.4), 6.0 (PDF 1.5), or 7.0 (PDF 1.6).
- Specifying other file and device settings such as resolution and optimization.

NOTE: If you create files with Acrobat 7.0 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 ["General Parameters" on page 56](#) for a description of each general parameter and the corresponding name in the UI.

Using the Image 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, JPEG2000, CCITTFax, RunLength, Flate, or automatic compression.

Distiller also checks to determine whether the existing clip path causes more than 10 percent of the image samples to fall outside the clip path. If this is the case, Distiller discards (crops) the image samples that fall outside the clip area, resulting in smaller images and PDF files. However, in some workflows, the full-size (non-cropped) images are extracted for special-purpose image manipulation, so Distiller provides parameters that can be used to disable cropping. See ["Disabling of Image Cropping" on page 41](#) for more information.

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”](#)

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 “[Flate Compression](#)” on [page 20](#) for details.

[Table 2.1, “Distiller settings for JPEG compression,”](#) shows an example of how Distiller sets the JPEG values through Distiller parameters.

TABLE 2.1 *Distiller settings for JPEG compression*

Distiller Parameter	Value Set
AutoFilterColorImages	false
AutoFilterGrayImages	false
EncodeColorImages	true
EncodeGrayImages	true
ColorImageFilter	DCTEncode
GrayImageFilter	DCTEncode
ColorImageDict	Medium (See Table 2.6, “Image compression quality.”)
GrayImageDict	Medium (See Table 2.6, “Image compression quality.”)

For automatic compression, **ColorACSDict** and **GrayACSDict** also need to be set. See “[Automatic Compression](#)” on [page 24](#) for more information.

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. 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.

Table 2.2, "Distiller settings for Flate compression," shows an example of how Distiller sets the Flate values through Distiller parameters.

TABLE 2.2 *Distiller settings for Flate compression*

Distiller Parameter	Value Set
AutoFilterColorImages	false
AutoFilterGrayImages	false
EncodeColorImages	true
EncodeGrayImages	true
EncodeMonolImages	true
ColorImageFilter	FlateEncode
GrayImageFilter	FlateEncode
MonolImageFilter	FlateEncode

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.

JPEG2000 Compression

JPEG2000 is a new international standard for the compression and packaging of image data. The JPEG2000 standard has features that make it attractive for general image compression in support of viewing. The JPEG2000 standard defines a wavelet-based method for image compression that gives somewhat better size reduction than other methods such as regular JPEG or CCITT. It is suitable both for images that have a single color component and for those with multiple color components.

The JPEG2000 compression algorithms are especially well suited for color images with smooth variation in color values. The JPEG2000 compression filter provides the ability to encode different versions of an image. These different versions form a progression of images with varying resolution. For example, a thumbnail version of the image may be encoded in the data followed by a sequence of other versions of the image, each with approximately 2x2 as many samples as the previous one. The last version is the highest resolution image corresponding to the value of the **Quality** key. This allows a PDF viewer to expend only the effort needed to decode the resolution that best matches the current viewing or printing needs. For example, if the full resolution version is highly sampled (e.g., 1200 pixels-per-inch), when displaying at actual size the PDF viewer can select and decode something less. Less time is spent decoding and fewer bytes have to be pushed through

the image pipeline. In addition, if one is viewing the file over the Web, not all of the image data need be downloaded. JPEG2000 data also has a built-in tiling structure which the PDF viewer can exploit. This means that if the viewer is zoomed in on a particular part of an image, only those tiles being displayed need to be decoded (to an appropriate resolution). Once again this is a performance win. Over the Web, this is particularly attractive when viewing and zooming in on high resolution images.

NOTE: The JPEG2000 compression filter is not available at the PostScript language level. There is no way in PostScript to invoke a JPEG2000Encode or JPEG2000Decode filter directly (there is no filter name defined for JPEG2000 compression in the PostScript language definition). For more information on PostScript language filters, see the *PostScript Language Reference, third edition*, section 3.13, Filtered Files Details.

Distiller provides four new image dictionaries for supporting JPEG2000 image compression. The new dictionaries are:

- **JPEG2000ColorImageDict**,
- **JPEG2000GrayImageDict**,
- **JPEG2000ColorACSIImageDict**, and
- **JPEG2000GrayACSIImageDict**.

These dictionaries are included in the default Adobe PDF settings files. They can only be used in Distiller's settings files in the dictionary associated with the **setdistillerparams** operator. They cannot be used at the PostScript language level in a PostScript file that is being processed by Distiller.

NOTE: Distiller silently accepts parameter values outside of the defined ranges in these dictionaries. Use caution when setting these values in custom Adobe PDF settings files.

Table 2.3, "Distiller settings for JPEG2000 compression," shows an example of how Distiller sets the JPEG2000 values through Distiller parameters.

TABLE 2.3 *Distiller settings for JPEG2000 compression*

Distiller Parameter	Value Set
AutoFilterColorImages	false
AutoFilterGrayImages	false
EncodeColorImages	true
EncodeGrayImages	true
ColorImageFilter	JPXEncode
GrayImageFilter	JPXEncode
JPEG2000ColorImageDict	See Table 2.4, "Distiller parameters for JPEG2000 image dictionaries."

TABLE 2.3 *Distiller settings for JPEG2000 compression*

Distiller Parameter	Value Set
JPEG2000GrayImageDict	See Table 2.4, “Distiller parameters for JPEG2000 image dictionaries.”

For automatic compression, **JPEG2000ColorACSIImageDict** and **JPEG2000GrayACSIImageDict** also need to be set. See [“Automatic Compression” on page 24](#) for more information.

The JPEG2000 dictionary parameters are shown in [Table 2.4, “Distiller parameters for JPEG2000 image dictionaries.”](#) Since all parameters are optional, an empty dictionary is acceptable.

NOTE: Distiller accepts invalid numeric values outside of the defined ranges for the values of the JPEG2000 image dictionaries. However, Distiller does generate an error if the values are of the wrong type.

TABLE 2.4 *Distiller parameters for JPEG2000 image dictionaries*

Key	Type	Value
TileWidth	integer	(Optional) The width of JPEG2000 image tiles in samples. Valid values are 128 - 2048 . If this key is absent, the default value is 256 . Values less than 128 or greater than 2048 will generate a range error.
TileHeight	integer	(Optional) The height of JPEG2000 image tiles in samples. Valid values are 128 - 2048 . If this key is absent, the default value is 256 . Values less than 128 or greater than 2048 will generate a range error.

TABLE 2.4 *Distiller parameters for JPEG2000 image dictionaries*

Key	Type	Value
Quality	integer	(Optional) The required image quality for the highest resolution image in the image progression. Valid values are 1,2,3,...100. 1 means lowest quality (highest compression), 99 means visually lossless compression, and 100 means numerically lossless compression. If this key is absent, the default value is 15 (Medium). In the Compression panel of the Distiller UI, the mapping that occurs for the predefined options is as follows: Minimum = 5 Low = 10 Medium = 15 High = 20 Maximum = 30 Lossless = 100

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 if producing PDF 1.4 or lower, and JPEG2000 if 1.5 or higher. When Distiller selects JPEG compression, it uses the parameters specified in either the **ColorACSIImageDict** or **GrayACSIImageDict** dictionary. For JPEG2000 images (only supported in PDF 1.5 and above), Distiller uses lossy JPEG2000 compression for low-frequency images, and lossless JPEG2000 compression for high-frequency images. When Distiller selects JPEG2000 compression, it uses the parameters specified in either the **JPEG2000ColorACSIImageDict** or **JPEG2000GrayACSIImageDict** dictionary.

Table 2.5, “Automatic Distiller parameters,” shows an example of how Distiller sets the automatic values through Distiller parameters. “[Decision Flow for Color and Grayscale Image Compression](#)” on page 37 describes how Distiller selects filters during automatic compression. “[Use of Image Dictionaries for Color and Grayscale Image Compression](#)” on page 38 describes how Distiller selects dictionaries during automatic compression.

TABLE 2.5 Automatic Distiller parameters

Distiller Parameter	Value Set
AutoFilterColorImages	true
AutoFilterGrayImages	true
EncodeColorImages	true
EncodeGrayImages	true
ColorImageAutoFilterStrategy	JPEG or JPEG2000
GrayImageAutoFilterStrategy	JPEG or JPEG2000
ColorACSIImageDict	Medium (See Table 2.6. .)
GrayACSIImageDict	Medium (See Table 2.6. .)
JPEG2000ColorACSIImageDict	See Table 2.4.
JPEG2000GrayACSIImageDict	See Table 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*.

The JPEG2000 filter compresses color or grayscale images using JPEG2000 compression. The JPEG2000 filter bases compression on settings contained in the dictionaries **JPEG2000ColorImageDict** and **JPEG2000GrayImageDict**, as described in “[JPEG2000 Compression](#)” on page 21.

Distiller Settings and Resulting Image Quality for the DCTEncode Filter

You can use the **Quality** field in the Distiller UI to set three parameters in the DCTEncode parameter dictionary to achieve five levels of image quality. Specifically, you can set 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**— Note that you can set this parameter individually in the Adobe PDF settings file, but the new value will be ignored.
- **VSamples**— Note that you can set this parameter individually in the Adobe PDF settings file, but the new value will be ignored.
- **QFactor**

For the DCTEncode filter, the only parameter that can be modified by the user, either via the UI or by editing the Adobe PDF settings file, is **QFactor**. From the **QFactor** value, Distiller sets the **HSamples** and **VSamples** arrays to hardcoded values as follows:

If **QFactor** ≥ 0.5 , then both the **HSamples** and **VSamples** arrays are hardcoded to **[2 1 1 2]**.

If **QFactor** < 0.5 , then both the **HSamples** and **VSamples** arrays are hardcoded to **[1 1 1 1]**.

The original values of the **HSamples** and **VSamples** arrays in the Adobe PDF settings file are completely ignored by Distiller. If you choose to save the settings to a file, Distiller saves the computed values for **HSamples** and **VSamples** in the file, regardless of the original values that were present in the file.

[Table 2.6, “Image compression quality”](#) summarizes the values Distiller sets to achieve Minimum, Low, Medium, High, and Maximum image quality.

TABLE 2.6 *Image compression quality*

Quality	HSamples	VSamples	QFactor
Minimum	[2 1 1 2]	[2 1 1 2]	2.40
Low	[2 1 1 2]	[2 1 1 2]	1.30
Medium	[2 1 1 2]	[2 1 1 2]	0.76
High	[1 1 1 1]	[1 1 1 1]	0.40
Maximum	[1 1 1 1]	[1 1 1 1]	0.15

The PostScript DCTEncode dictionary includes several parameters that cannot be set through the Distiller DCTEncode dictionary. These unsettable parameters include **Columns**, **Rows**, **Colors**, **QuantTables**, **HuffTables**, **ColorTransform**, and **CloseTarget**. These parameters are set internally in Distiller depending on the properties of each image.

The **ColorTransform** parameter is set by Distiller to the “best” value for each image. (See the description of the DCTEncode parameter dictionary in Section 3.13.3 in the *PostScript Language Reference, third edition*.) Internally the algorithm is:

ColorTransform is set to 0 if the colorspace is Lab or Gray or (cmyk AND **QFactor** ≥ 0.5). Otherwise, **ColorTransform** is set to 1.

The internal **Blend** parameter is always set to 1.

Customizing Color and Grayscale Image Compression for the DCTEncode Filter

Distiller provides “Distiller-specific” parameters for customizing compression values for the DCTEncode filter 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:

- **ColorACSIImageDict**
- **ColorImageDict**

- **GrayACSIImageDict**
- **GrayImageDict**

You can customize color or grayscale image compression for the DCTEncode filter by setting values for the **QFactor** dictionary key, as described above. Distiller applies the compression you choose to all colors or grayscale bitmap images in a PDF file. See “[Controlling Downsampling and Encoding for Each Sampled Image](#)” on page 32 for details on controlling compression of individual images in a file.

NOTE: When either **AutoFilterColorImages** or **AutoFilterGrayImages** is **true**, Distiller looks at the image data to determine whether to use JPEG or Flate compression. If Distiller decides to use JPEG compression, it uses the JPEG parameters in the **ColorACSIImageDict** and **GrayACSIImageDict**. If, on the other hand, **AutoFilterImages** is **false**, and either **ColorImageFilter** or **GrayImageFilter** is set to JPEG (DCTEncode), Distiller uses the JPEG parameters in **ColorImageDict** and **GrayImageDict**. This explains the difference between **ColorACSIImageDict** and **ColorImageDict**, and **GrayACSIImageDict** and **GrayImageDict**.

Customizing Color and Grayscale Image Compression for the JPEG2000 Filter

Distiller provides “Distiller-specific” parameters for customizing compression values for the JPEG2000 filter for color and grayscale images. Each of the Distiller parameters below is of type dictionary and contains keys for customizing the JPEG2000 filter:

- **JPEG2000ColorACSIImageDict**
- **JPEG2000ColorImageDict**
- **JPEG2000GrayACSIImageDict**
- **JPEG2000GrayImageDict**

You can customize color or grayscale image compression for the JPEG2000 filter by setting your own values for any of the JPEG2000 dictionary keys in the dictionaries listed above. For a list of the available keys, see [Table 2.4, “Distiller parameters for JPEG2000 image dictionaries.”](#) Distiller applies the compression you choose to all colors or grayscale bitmap images in a PDF file. See “[Controlling Downsampling and Encoding for Each Sampled Image](#)” on page 32 for details on controlling compression of individual images in a file.

Setting Compression of Monochrome (Black and White) Images

Distiller uses a different selection of compression filters for monochrome image compression. The available filters are: CCITTFaxEncode, RunLengthEncode, and FlateEncode.

NOTE: With the exceptions of the **AntiAliasMonolImages** and **MonolImageDepth** 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*.

Customizing Monochrome Image Compression for the CCITTFax Filter

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 **MonolImageDict** 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 **MonolImageDict** parameter to customize monochrome image compression.

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 of the *PostScript Language Reference, third edition*.

NOTE: The JPEG2000 compression filter is not available at the PostScript language level. There is not a way in PostScript of invoking a JPXEncode or JPXDecode filter directly (there is no filter name defined for JPEG2000 compression in the PostScript language definition). For more information on PostScript language filters, see the *PostScript Language Reference, third edition*, Section 3.13, Filtered Files Details. There is no information about the JPEG2000 filter in the *PostScript Language Reference, third edition*. All the information about the JPEG2000 filter is in this document. For information about JPEG2000 encoding, see <http://www.jpeg.org/JPEG2000.htm>.

Downsampling and Subsampling Images

Distiller can reduce the number of pixels-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 pixels-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 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 pixels-per-inch to a typical monitor resolution of 72 pixels-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**
- **MonolImageDownsampleThreshold**

The downsampling threshold is the ratio of the input image resolution (specified by the value of the respective **ColorImageResolution**, **GrayImageResolution**, or **MonolImageResolution** 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 pixels-per-inch:

```
trunc((72 * 1.5) + .5) = 108 pixels-per-inch
```

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 1.5.

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.

Controlling the Range of Bit Depths For Which Downsampling Occurs

Distiller allows you to control the range of bit depths for which downsampling occurs. For example, in a workflow where there is a mixture of 1-bit and 8-bit data, you can

downsample the 8-bit data while not touching the 1-bit data. This is done with the Distiller parameters shown in the following table:

TABLE 2.7 *Image compression quality*

Parameter	Type	Valid Values
ColorImageMinDownsampleDepth	integer	1, 2, 4 or 8
GrayImageMinDownsampleDepth	integer	2, 4 or 8

For example, **/ColorImageMinDownsampleDepth 4** means that if **/DownsampleColorImages** is **true**, then Distiller only downsamples color images for which the bit depth is at least 4 bits-per-sample (i.e., only 4 and 8 bits-per-sample images).

NOTE: 12 bits-per-sample images (valid in PostScript) are treated exactly as 8 bits-per-sample images because they are converted to 8 bits-per-sample before downsampling takes place.

Similarly, **/GrayImageMinDownsampleDepth 8** means that if **/DownsampleGrayImages** is **true**, then Distiller only downsamples gray images for which the bit depth is 8 bits-per-sample.

As a final example, **/ColorImageMinDownsampleDepth 2** means that if **/DownsampleColorImages** is **true**, then Distiller only downsamples color images for which the bit depth is at least 2 bits-per-sample.

Specifying a Lower Limit for the Resolution of Sampled Images

During normal operation, Distiller provides a mechanism for limiting the resolution of sampled images placed in the PDF files. This mechanism is controlled by the downsampling parameters:

- **DownsampleColorImages**
- **ColorImageResolution**
- **ColorImageDownsampleThreshold**
- **DownsampleGrayImages**
- **GrayImageResolution**
- **GrayImageDownsampleThreshold**
- **DownsampleMonolImages**
- **MonolImageResolution**
- **MonolImageDownsampleThreshold**

The settings of these parameters guarantee that all images placed in the PDF file have a resolution that is equal to or smaller than **[Color|Gray|Mono]ImageResolution** multiplied by

[Color|Gray|Mono]DownsampleThreshold, as described in [, “Setting The Threshold Resolution For Downsampling Images” on page 29](#).

In addition, starting with version 7, Distiller is able to flag PDF files that contain images whose resolution don't meet a minimum requirement. An example is a sampled image in an advertisement where the image must be of a certain quality (i.e., must have a resolution higher than a minimum value). To provide this mechanism, the following six parameters are used:

TABLE 2.8 MinResolution Parameters

Parameter	Type	Valid Values
ColorImageMinResolution	integer	9 to 64000
ColorImageMinResolutionPolicy	name	OK, Warning, Error
GrayImageMinResolution	integer	9 to 64000
GrayImageMinResolutionPolicy	name	OK, Warning, Error
MonolImageMinResolution	integer	9 to 64000
MonolImageMinResolutionPolicy	name	OK, Warning, Error

The default values for these parameters in the predefined Adobe PDF settings files are chosen to be the same as the values for the default downsampling resolution. With these default values, Distiller's default behavior does not change (i.e., Distiller does not enforce any lower limit on image resolution).

Whereas the downsampling parameters ensure that the user can limit the resolution (and the size of the resulting file) for images that go into the PDF, the MinResolution parameters ensure that a PDF file will not have any images with lower resolution than the defined limit. This feature is thus primarily for prepress people who want to detect that no low resolution images make it into into a PDF file.

The two sets of parameters guard each end of the spectrum of image resolution. At the high end, Distiller actively ensures that images don't have too high a resolution by downsampling them. At the low end, Distiller has a passive check (the **[Color|Gray|Mono]MinResolutionPolicy**) which can ignore, warn, or provide an error message if a low resolution image is about to be placed into the PDF. Distiller cannot actively alter these images because it doesn't support up-sampling.

If you get a warning/error about a low resolution image and you have set the MinResolution parameters correctly according to your requirements, there is really nothing you can do in Distiller to get a higher resolution image. You need to go to the source of the image and regenerate it with a higher resolution.

How to Limit Image Resolution

The following is an example of how to use the new parameters to limit image resolution:

If you want Distiller to issue a warning every time a sampled gray image with resolution smaller than 100 ppi is placed in the PDF file, then set these two parameters:

```
/GrayImageMinResolution 100
/GrayImageMinResolutionPolicy /Warning
```

The warning messages will look like this:

**%%[Warning: Gray image resolution (92 ppi) is
lower than /GrayImageMinResolution (100 ppi)]%%**

If **GrayImageMinResolutionPolicy** is set to **/Error**, then an error message is emitted and the job fails with a limitcheck error. If **GrayImageMinResolutionPolicy** is set to **/OK**, then Distillation continues normally.

There are no dependencies or interaction between the new “MinResolution” parameters and the downsampling parameters.

Normally you would not set the resolution policy to **/Warning** or **/Error** and at the same time set the minimum resolution to a value that is higher than the downsampling threshold. If you do this, the result is that *all* images are flagged as having too low a resolution. This setting could be used to produce PDF files that have no sampled images in them whatsoever.

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.

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 pixels-per-inch monochrome image that is downsampled to 150 pixels-per-inch. 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

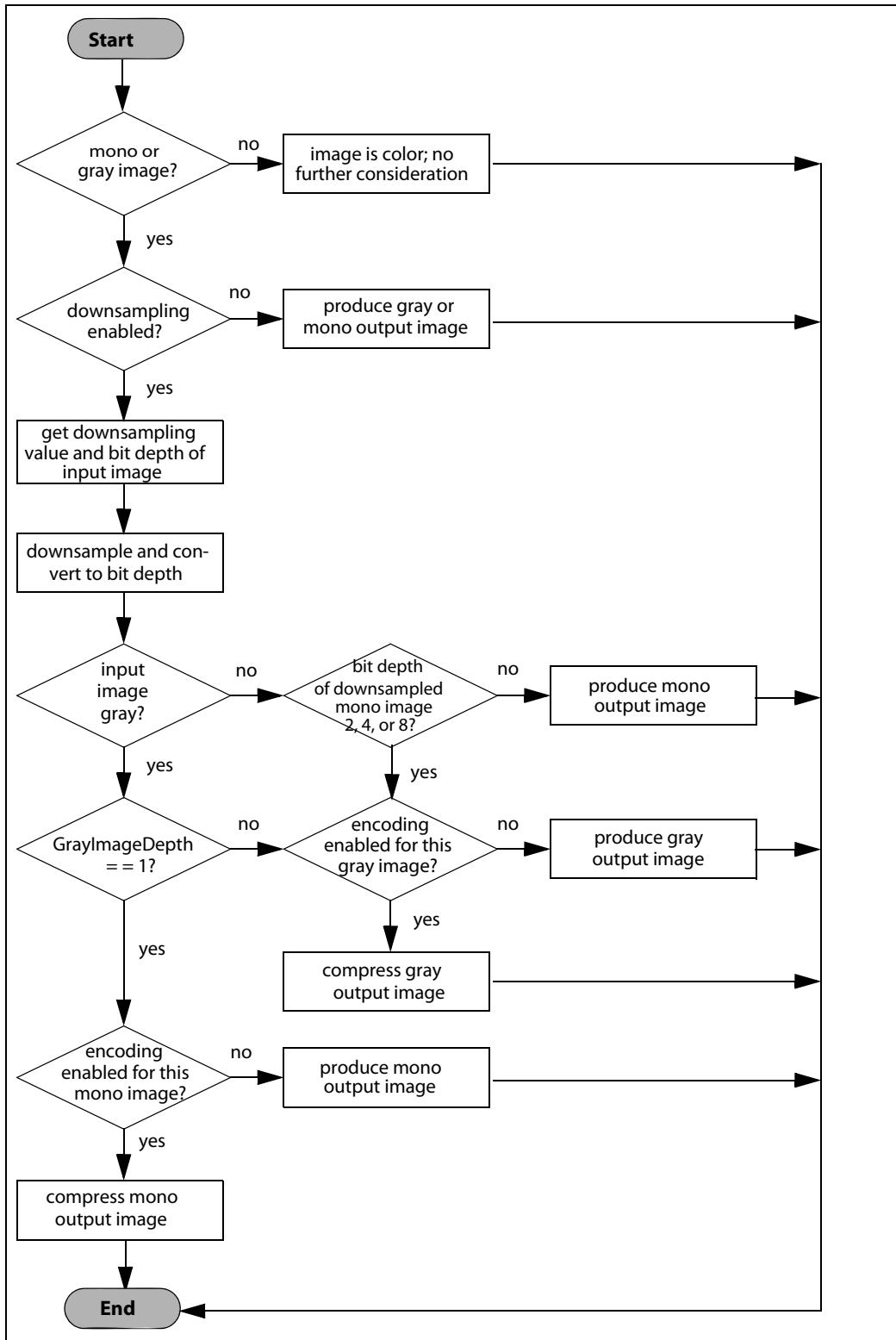
five values, the PDF file format only supports bit depths of 1, 2, 4, and 8 and hence the data are encoded using 4 bits-per-sample.

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 **MonolImageDepth** 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 image 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.

Overview of Distiller's Filter Selection Strategy for Images

Distiller uses the following criteria for selecting a filter for compressing images.

When **AutoFilterColorImages** (or **AutoFilterGrayImages**) is set to **true**, Distiller selects lossless FlateEncode (for gray and color images) if:

1. The image is DeviceN, or
2. The image is small (< 1024 bytes), or
3. The image is ChromaKeyed, or
4. The samples are not smooth.

Otherwise, Distiller selects the DCTEncode filter.

When **AutoFilterColorImages** (or **AutoFilterGrayImages**) is set to **false**:

- For Monochrome images (1 color, 1 bpc) Distiller allows these filters:
 - CCITTFaxEncode, CCITT Group 3 (/K=0)
 - CCITTFaxEncode, CCITT Group 4 (/K=-1)
 - RunLengthEncode
 - FlateEncode
- For Grayscale images (1 color, 2, 4 bpc) Distiller allows this filter:
 - FlateEncode
- For Grayscale images (1 color, 8 bpc) Distiller allows these filters:

- JPXEncode
- DCTEncode
- FlateEncode
- For Grayscale images (1 color, 12 bpc) Distiller removes the least significant four bits of each image sample, yielding 8 bits-per-sample. Distiller then allows these filters:
 - JPXEncode
 - DCTEncode
 - FlateEncode
- For Color images (3, 4 colors, 1, 2, 4 bpc) Distiller allows this filter:
 - FlateEncode
- For Color images (3, 4 colors, 8 bpc) Distiller allows these filters:
 - JPXEncode
 - DCTEncode
 - FlateEncode
- For Color images (3, 4 colors, 12 bpc) Distiller removes the least significant four bits of each image sample, yielding 8 bits-per-color. Distiller then allows these filters:
 - JPXEncode
 - DCTEncode
 - FlateEncode

When image compression is selected (with **EncodeColorImages**, **EncodeGrayImages**, or **EncodeMonolImages**), but no filter name is specified (**ColorImageFilter**, **GrayImageFilter**, or **MonolImageFilter** is absent), then Distiller selects lossless FlateEncode in all cases. The following filters are *never* selected, even if they are specified in the Adobe PDF settings file:

- LZWEncode
- ASCII85Encode
- ASCIIHexEncode

If an invalid filtername is specified, (for example, **/ColorImageFilter /DCTEncode27**), then Distiller emits the error message:

/ColorImageFilter out of range

Finally, Distiller also performs a filter “fallback.” For the following conditions, Distiller “falls back” to FlateEncode:

1. The selected filter is CCITTFaxEncode and the image is wide (more than 40,000 columns).
2. The selected filter is JPXEncode and the image is indexed or ChromaKeyed or the compatibilityLevel is less than 1.5.
3. The selected filter is DCTEncode and the image is wide (more than 40,000 columns), indexed, deviceN or ChromaKeyed.
4. If the selected filter is not supported for the number of colors or sample depth of the particular image being compressed.

Decision Flow for Color and Grayscale Image Compression

Distiller's decision flow for color and grayscale image compression is expressed in the following pseudo-code:

```
if (/Encode [Color|Gray] Images == true)
{
    if (/AutoFilter[Color|Gray] Images == true)
    {
        if (CompatibilityLevel >= 1.5) and
        (/ [Color|Gray] ImageAutoFilterStrategy == /JPEG2000)
        {
            If (low freq)
                Use lossy JPEG2000
            else
                Use lossless JPEG2000
        }
        else
        {
            if (low freq)
                Use lossy JPEG
            else
                Use lossless Flate
        }
    }
    else
    {
        Use the compression filter specified
        in / [Color|Gray] ImageFilter
    }
}
else
{
    Don't compress images
}
```

NOTE: When Distiller uses lossy JPEG2000 compression for low-frequency images, the compression parameters in **JPEG2000ColorACSDict** or **JPEG2000GrayACSDict** are used. When Distiller uses lossless JPEG2000 compression for high-frequency images, the compression parameters in **JPEG2000ColorACSDict** or **JPEG2000GrayACSDict** are used with the modification that the **Quality** parameter is forced to 100 (to achieve lossless compression). See ["Use of Image Dictionaries for Color and Grayscale Image Compression" on page 38](#).

NOTE: When Distiller uses lossy JPEG compression for low-frequency images, the compression parameters in **ColorACSDict** or **GrayACSDict** are used. When Distiller uses lossless Flate compression for high-frequency images, no dictionary parameters are used (because Flate does not require any parameters). See [Use of Image Dictionaries for Color and Grayscale Image Compression](#), below.

Use of Image Dictionaries for Color and Grayscale Image Compression

Distiller's use of image dictionaries for color and grayscale compression is expressed in the following pseudo-code:

```

if (/AutoFilter[Color|Gray] Images == true)
{
    Use these dictionaries:
        /ColorACSIImageDict
        /GrayACSIImageDict
        /JPEG2000ColorACSIImageDict
        /JPEG2000GrayACSIImageDict
}
else
{
    Use these dictionaries:
        /ColorImageDict
        /GrayImageDict
        /JPEG2000ColorImageDict
        /JPEG2000GrayImageDict
}

```

Compression Parameters Example

The PostScript example below shows a code fragment specifying that monochrome images be downsampled to 72 pixels-per-inch, 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-pixels-per-inch image, it is downsampled to 75 pixels-per-inch, the closest possible value to the 72 pixels-per-inch requested.

```

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

```

Setting Compression of Text, Line Art, and Objects

You can use the **CompressPages** parameter to set the compression of text and line art. For PDF 1.5 and above, you can use the **CompressObjects** parameter to control object-level compression, which is the consolidation of small objects (each of which isn't compressible itself) into *streams* which can then be efficiently compressed.

Passing Through JPEG Images

The **PassThroughJPEGImages** parameter is used to pass JPEG images (images that are already compressed with the DCTEncode filter) through Distiller without recompressing them. If this parameter is set to **true**, Distiller will decompress JPEG images to ensure that they are not corrupt, but it will not recompress valid images, passing the original image through untouched.

The advantages to this are the following:

- Performance — Only decompression and not recompression occurs.
- No loss of image data — DCT encoding inherently causes some loss of data; thus, with this option, since no recompression occurs, no data is lost.
- No loss of metadata — When Distiller decompresses an image, all metadata is discarded; thus, with this option, no metadata is lost since no recompression on the decompressed image occurs.

The default value for this parameter is **true**. Normally there is no reason to disable the passthrough feature. The following describes exactly how the parameter is interpreted.

If **true**:

Attempt to pass through all compressed JPEG images. Compress images that are not compressed according to the filter selected by these Adobe PDF parameters:

- **EncodeColorImages**
- **ColorImageFilter**
- **AutoFilterColorImages**
- **ColorImageAutoFilterStrategy**
- **ColorACSIImageDict**
- **ColorImageDict**
- **JPEG2000ColorACSIImageDict**
- **JPEG2000ColorImageDict**
- **EncodeGrayImages**
- **GrayImageFilter**
- **AutoFilterGrayImages**
- **GrayImageAutoFilterStrategy**
- **GrayACSIImageDict**
- **GrayImageDict**
- **JPEG2000GrayACSIImageDict**
- **JPEG2000GrayImageDict**
- **EncodeMonolImages**
- **MonolImageFilter**

- **MonolImageDict**

If **false**:

Decompress all images and recompress them according to the values of the above list of Adobe PDF parameters.

It should also be noted that Distiller allows JPEG passthrough only in the following circumstances:

- The image passes the “downsampling” test—i.e., it will not be downsampled. JPEG passthrough is automatically turned off when an image is a candidate for downsampling.
- **ColorConversionStrategy** is not **sRGB**.
- **ColorConversionStrategy** is **sRGB** and current PostScript color space (for the image) is **DeviceRGB** or **DeviceGray**.
- The image will not be cropped—i.e., the clippath is such that less than 10% (or none) of the image pixels will be removed.

NOTE: The Smallest File Size predefined Adobe PDF settings file sets **PassThroughJPEGImages** to **false**. This generally results in smaller file sizes, though there are cases where this setting could actually increase file size. This would happen, for example, if the original JPEG in the PostScript file was compressed with a very low **Quality** setting—lower than the **Quality** setting in the Smallest File Size file.

Summary of Distiller Image Compression Controls

The following table summarizes Distiller’s image compression controls and how they are used.

TABLE 2.9 Image Compression Controls Summary

Control	Description
ColorImageFilter	Specify which compression filter to use in non-automatic compression mode.
GrayImageFilter	
AutoFilterColorImages	Hold boolean values that enable automatic compression.
AutoFilterGrayImages	
ColorImageAutoFilterStrategy	Determine whether to use JPEG + FLATE or JPEG2000.
GrayImageAutoFilterStrategy	
ColorACSIImageDict	Dictionaries used for JPEG compression in automatic compression mode.
GrayACSIImageDict	

TABLE 2.9 *Image Compression Controls Summary*

Control	Description
ColorImageDict	Dictionaries used for JPEG compression in regular (non-automatic) compression mode.
GrayImageDict	
JPEG2000ColorACSIImageDict	Dictionaries used for JPEG2000 compression in automatic compression mode.
JPEG2000GrayACSIImageDict	
JPEG2000ColorImageDict	Dictionaries used for JPEG2000 compression in regular (non-automatic) compression mode.
JPEG2000GrayImageDict	

Disabling of Image Cropping

Distiller checks to determine whether the existing clip path causes more than 10 percent of the image sample to fall outside the clip path. If this is the case, Distiller discards (crops) the image sample, resulting in smaller images and PDF files.

However, in some workflows, the full-size (non-cropped) images are extracted for special-purpose image manipulation. To disable cropping, the parameters listed and described in [Table 2.10, “Parameters Used to Disable Cropping,”](#) are used. **false** indicates that Distiller should not clip image samples regardless of the current clip area. **true** indicates that Distiller should crop only if the 10 percent criteria is met.

TABLE 2.10 *Parameters Used to Disable Cropping*

Parameter	Type	Default
CropColorImages	boolean	true
CropGrayImages	boolean	true
CropMonolImages	boolean	true

The default for all three is **true**, which is the normal behavior for Distiller.

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) for the characters used in the document. Subsetting is expressed as a percentage of the font glyphs for a font format. The **SubsetFonts** and **MaxSubsetPct** parameters are used to control partial embedding of fonts.

Table 2.11, “Distiller control over embedding and subsetting fonts,” identifies the fonts that you can (or cannot) embed or subset through Distiller parameters.

TABLE 2.11 *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
OpenType	Yes	Yes	Yes

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](#).

NOTE: Distiller 5 and above also support 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.

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.

NOTE: A font may not be embedded if its license doesn’t permit embedding, even though its name is in the **AlwaysEmbed** list or **EmbedAllFonts** is **true**. Furthermore, a symbolic font is always embedded (if license permits) even if its name is in the **NeverEmbed** list.

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’.

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 Section 4.5.4 of the *PDF Reference* for details on profiles.

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 UI.
- **UseDeviceIndependentColor**. This is equivalent to “Tag Everything for Color Management” in the Distiller UI.
- **UseDeviceIndependentColorForImages**. This is equivalent to “Tag Only Images for Color Management” in the Distiller UI.
- **sRGB**. This is equivalent to “Convert All Colors to sRGB” in the Distiller UI.
- **CMYK**. This is equivalent to “Convert All Colors to CMYK” in the Distiller UI.

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

Table 2.12, “PS color space (in) vs. PDF color space (out),” shows how Distiller converts the PostScript input to the equivalent color space for each of the **ColorConversionStrategy** parameter values.

In Table 2.12, X/Y entries mean “X in PDF 1.2 and Y in PDF 1.3 or higher.”

TABLE 2.12 PS color space (in) vs. PDF color space (out)

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

TABLE 2.12 PS color space (in) vs. PDF color space (out)

PS Input	LeaveColorUnchanged	UseDeviceIndependentColor	UseDeviceIndependentColorForImages	sRGB	CMYK
RGB image	RGB	ICCBased	CalRGB/ICCBased	CalRGB/ sRGB	CMYK
CMYK text and graphics	CMYK	LAB/ICCBased	CMYK	CalRGB/ sRGB	CMYK
CMYK image	CMYK	LAB/ICCBased	LAB/ICCBased	CalRGB/ sRGB	CMYK
CIE text and graphics	Cal/ICC	Cal/ICC	Cal/ICC	CalRGB/ sRGB	Gray or CMYK ^a
CIE image	Cal/ICC	Cal/ICC	Cal/ICC	CalRGB/ sRGB	Gray or CMYK ^a

a. CIEBasedA becomes Gray, others become CMYK

Using the Advanced Adobe PDF Parameters

Advanced Adobe PDF settings 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
- Controlling EPS, OPI, and DSC information
- Allowing transparency
- Creating Job Definition Format (JDF) files
- Embedding of Adobe PDF settings

See [Table 3.8, “Advanced parameters,”](#) for a description of each of the advanced Distiller parameters. Six of the advanced Adobe PDF settings discussed in this section are:

- **DetectBlends**
- **ParseDSCCommentsForDocInfo**
- **PreserveEPSInfo**
- **CreateJobTicket**
- **AllowTransparency**

- **CreateJDFFile**

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) was controlled by the **IdiomRecognition** PostScript feature, while the second method was controlled by **DetectBlends**. You had to turn off **IdiomRecognition** to use **DetectBlends**.

In Distiller 5.0 and above, **DetectBlends** controls the blend detecting idioms. By default **IdiomRecognition** is turned on in Distiller 5.0 and above, and the blend detecting idioms are controlled using the **DetectBlends** parameter. You can still use the PostScript feature **IdiomRecognition** with the **setuserparams** operator, if needed.

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 2.13, “Document information,”](#) lists this information.

TABLE 2.13 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 7.0")
CreationDate	from Distiller time stamp (creation time of PDF file)

TABLE 2.13 Document information

Document Information	Source
ModDate	from Distiller time stamp (creation time of PDF file)

Starting with version 5, Distiller 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 Adobe's Extensible Metadata Platform (XMP).

PreserveEPSInfo Parameter Details

If **PreserveEPSInfo** is **true**, Distiller attempts to preserve the encapsulated PostScript (EPS) information as properties of the PDF document. [Table 2.14, "EPS information,"](#) lists this information.

TABLE 2.14 EPS information

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

Starting with version 5, Distiller 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 Adobe's XMP.

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 2.15, “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 2.15, “Relationship between **setpagedevice** keys and job ticket keys,” “First” identifies **setpagedevice** keys that must appear on the first page.

TABLE 2.15 *Relationship between **setpagedevice** keys and job ticket keys*

setpagedevice key	PS page	job ticket key
Bind	First	JobTicketContents::Finishing
NOTE: Bind is unrelated to the Distiller parameter Binding .		
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 <i>Portable Job Ticket Format, Version 1.1</i> , 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

TABLE 2.15 Relationship between *setpagedevice* keys and job ticket keys

setpagedevice key	PS page	job ticket key
MediaType	First	JobTicketContents::Media::Category
MediaWeight	First	JobTicketContents::Media::Weight
MirrorPrint	First	JobTicketContents::MediaUsage::MirrorPrint
NegativePrint	First	JobTicketContents::MediaUsage::NegativePrint
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 <i>setpagedevice</i> 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.

CreateJDFFile Parameter Details

The Job Definition Format (JDF) Specification is owned and maintained by the International Cooperation for the Integration of Processes in Prepress, Press and PostPress (CIP4) (www.cip4.org). Distiller 7 complies with JDF Specification Version 1.1 Revision A, published on September 5, 2002. It is available on the web at:

http://www.cip4.org/documents/jdf_specifications/index.html

NOTE: The Acrobat 7 Professional product now supports creation of both JDF 1.1- and JDF 1.2-compliant JDF files. For more information, see the *Acrobat Guide* in Distiller online Help.

If the **CreateJDFFile** parameter, is set to **true**, Distiller produces a Job Definition Format (JDF) file that reflects the parameters used for distillation. The Adobe Normalizer product (see *Using Adobe Normalizer Server, Version 6.0.4*) is also capable of producing JDF files, but it can consume them as well. [Appendix A](#) describes how Normalizer interprets and converts Distiller parameters; use this information to understand the JDF file created by Distiller.

The JDF file is output to the current directory with the **.jdf** extension. The filename is the same as the **.log** file and the file that is being distilled. (The “current directory” is the directory where the new PDF file is output.)

pdfmark for JDF

The **JDF** pdfmark allows the PostScript file/stream being distilled to specify certain elements and attributes to be added to a JDF file. For details, see *Using Adobe Normalizer Server, Version 6.0.4* and *pdfmark Reference Manual*.

Using the Standards Parameters

The Standards parameters provide control over PDF/A- and PDF/X-compliant output. PDF/A is a proposed ISO standard for the long-term preservation (archival) of electronic documents. PDF/X is a focused subset of PDF designed specifically for reliable prepress data interchange and it is an International Standards Organization (ISO) standard (www.iso.org). The PDF/X parameters are supported only in the Adobe Acrobat 6.0 (and above) Professional version of the product. Jobs distilled using these settings will fail if the data submitted to Distiller doesn't comply with the requirements for the specific version of the ISO specification. For additional information, see the *Acrobat Guide* in Distiller online Help.

Distiller checks for compliance with PDF/A and PDF/X standards using the **CheckCompliance**, **PDFX1aCheck**, and **PDFX3Check** parameters. There are also several parameters used to control the **TrimBox**, **BleedBox**, **ArtBox**, and **MediaBox** entries for page objects. These and other PDF/X parameters are used in providing content when creating an output intent dictionary if a PostScript document does not provide the required information (see Section 10.10 of the *PDF Reference* for details). A summary of the Standards parameters is found in [“Standards Parameters \(Acrobat Professional Only\)” on page 100](#).

Using the **CheckCompliance**, **PDFX1aCheck**, and **PDFX3Check** Parameters

To enable Distiller 7 settings files to be used with Distiller 6, the parameters:

`/PDFX1aCheck true`

and

```
/PDFX3Check true
```

are preserved and have the same meaning as:

```
/CheckCompliance [ /PDFX1a:2001 ]
```

and

```
/CheckCompliance [ /PDFX3:2002 ]
```

respectively. The **CheckCompliance** parameter, if present, overrides the **PDFX1aCheck** and **PDFX3Check** parameters.

If Neither PDFX1aCheck Nor PDFX3Check Is Present

If a settings file with only a **CheckCompliance** parameter (i.e., no **PDFX1aCheck** or **PDFX3Check**) is encountered and written out, the **PDFX1aCheck** and **PDFX3Check** parameters are written out with appropriate boolean values. Doing this enables backward compatibility with Distiller 6 for testing of PDF/X-1a:2001 and PDF/X-3:2002 standards compliance. That is,

```
/CheckCompliance [ /PDFX1a:2001 ]
```

without **PDFX1aCheck** and **PDFX3Check** is written out and also generates:

```
/PDFX1aCheck true
/PDFX3Check false
```

Similarly:

```
/CheckCompliance [ /PDFX3:2002 ]
```

without **PDFX1aCheck** and **PDFX3Check** is written out and also generates:

```
/PDFX1aCheck false
/PDFX3Check true
```

Any other values for **CheckCompliance** also generates:

```
/PDFX1aCheck false
/PDFX3Check false
```

If PDFX1aCheck and/or PDFX3Check Are Present

If either of these parameters is present, or both, then the rewriting of the file will not cause them to be changed. This, again, is to enable the Adobe settings file with the ability to be used both by Distiller 6 and Distiller 7.

If CheckCompliance is Missing

If the **CheckCompliance** parameter is missing, then it takes its value from the **PDFX1aCheck** parameter, if present and **true** (in which case the value for **CheckCompliance** is **[/PDFX1a:2001]**). If the **PDFX1aCheck** parameter is missing or **false**, then the **PDFX3Check** parameter is inspected.

If the **PDFX3Check** parameter is **true**, the the value for **CheckCompliance** is set as **[/PDFX3:2002]**. If none of these conditions apply, then the value for **CheckCompliance** is taken to be **[/None]**.

To summarize, the **CheckCompliance** parameter takes precedence over the **PDFX1aCheck** and **PDFX3Check** parameters by Distiller 7. For backward compatibility, the **PDFX1aCheck** and **PDFX3Check** parameters are either preserved or set to values consistent with that of the **CheckCompliance** parameter.

Using the **PDFXOutputIntentProfile** Parameter

On successful PDF/X validation, and when the PostScript document does not specify the output intent destination profile, the values defined by **PDFXOutputIntentProfile** and **PDFXOutputConditionIdentifier** are used.

The **PDFXOutputIntentProfile** parameter is used to identify a profile name. If the value of **PDFXOutputIntentProfile** is set to **(None)** or the empty string **()**, then the PostScript document must specify the output intent destination profile for PDF/X validation to succeed. If the value of **PDFXOutputIntentProfile** is set to **(Use Output Condition Identifier)**, then the value defined by the **PDFXOutputConditionIdentifier** is taken to satisfy the PDF/X requirement that an output intent destination profile be specified. Otherwise, the value of **PDFXOutputIntentProfile** is taken to be the output intent destination profile name. In the latter case, if the profile is present on the computer performing distillation, then that profile is embedded on successful distillation.

Examples

Example 1: The **PDFXOutputIntentProfile** is set to **(Euroscale Uncoated v2)** and Distiller knows the corresponding Output Condition Identifier:

```
12 0 obj
<<
/Type /OutputIntent
/S /GTS_PDFX
/OutputConditionIdentifier (FOGRA4)
/RegistryName (http://www.color.org)
/Info (Euroscale Uncoated v2)
/DestOutputProfile 11 0 R
>>
endobj
11 0 obj
<<
/N 4
/Length 388226
/Filter /FlateDecode
>>
stream
... ICCProfile data ...
endstream
endobj
```

Example 2: The **PDFXOutputIntentProfile** is set to (U.S. Web Uncoated v2) and Distiller does not know the corresponding Output Condition Identifier:

```
12 0 obj
<<
/Type /OutputIntent
/S /GTS_PDFX
/OutputConditionIdentifier (U.S. Web Uncoated v2)
/Info (U.S. Web Uncoated v2)
/DestOutputProfile 11 0 R
>>
endobj
11 0 obj
<<
/N 4
/Length 386435
/Filter /FlateDecode
>>
stream
... ICCProfile data ...
endstream
endobj
```

Example 3: The **PDFXOutputIntentProfile** is set to (Use Output Condition Identifier) and **PDFXOutputConditionIdentifier** is set to (myOutputConditionIdentifier):

```
11 0 obj
<<
/Type /OutputIntent
/S /GTS_PDFX
/OutputConditionIdentifier (myOutputConditionIdentifier)
>>
endobj
```


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

The “Default Value” column contains the default Adobe PDF settings that are “hard-coded” into Distiller. These are the options that are used if an option is not specified in the `.joboptions` file, or the `.joboptions` file is empty. These are the same as those in the Standard `.joboptions` file, with the following exceptions:

- **NeverEmbed** defaults to `[true]` (no list of fonts)
- **Description** is not provided
- **TransferFunctionInfo** defaults to `Preserve`
- **CompressObjects** defaults to `Off`
- **CalGrayProfile** defaults to `()`
- **PassThroughJPEGImages** defaults to `false`

General Parameters

Table 3.1, “General parameters,” 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	“Factory” Default Value	Semantics
AutoRotatePages Auto-Rotate Pages: Off, Individually, Collectively by File	name	All	Allows Distiller to automatically orient (rotate) pages based on the predominant text orientation. Auto-rotation is not done if the file contains the <code>%%ViewingOrientation</code> DSC comment and <code>ParseDSCComments</code> is true . If AutoRotatePages is set to None , pages are not automatically oriented and the <code>%%ViewingOrientation</code> DSC comment is ignored (even if AutoRotatePages is true). 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.
Binding Binding	name	Left	Controls the value of the PageDirection key in the <code>ViewerPreferences</code> 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	1.4	The PDF version number: 1.2, 1.3, 1.4, 1.5, or 1.6.

TABLE 3.1 General parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
CompressObjects Object-Level Compression: Off, Tags Only, Maximum NOTE: In the UI, the Maximum selection will appear only if CompatibilityLevel is set to 1.5 or above.	name	Off	<p>NOTE: This parameter must be present in the Adobe PDF settings file to take effect.</p> <p>Introduced with PDF 1.5. Controls object-level compression, which is the consolidation of small objects (each of which isn't compressible itself) into <i>streams</i> which can then be efficiently compressed. A new method of referencing these objects had to be invented, causing a backward compatibility issue.</p> <p>Off PDF 1.5 object streams and cross reference streams will not be used.</p> <p>Tags PDF 1.5 object streams feature may be used to compress tags. The compressed tag information will no longer be usable in Acrobat 5 or earlier Adobe Readers. The PDF files created using this setting will be openable and usable in Acrobat 5 and earlier Adobe Readers even though the tag information is not usable by Acrobat 5 and earlier Adobe Readers. Acrobat 6 (and above) Adobe Readers will, however, have full access to the tag information.</p> <p>All Maximum compression. Objects streams and cross reference streams may be used to reduce PDF file size. The compressed file will only be readable by Acrobat 6 (and above).</p>

TABLE 3.1 General parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
CoreDistVersion	integer	7000	<p>NOTE: This parameter does not appear in the UI.</p> <p>(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 UI.</p>
Description Description	dictionary	Absent	<p>Allows you to set language-dependent strings that are displayed when the containing Adobe PDF settings file is selected as active. The dictionary contains keys whose values are the standard Acrobat 3-letter codes (see the <i>Acrobat Core API Reference</i> for a listing of these codes). The value associated with each language key is a string that is intended to contain the description of the Adobe PDF settings file. It is assumed that the string will be reflowed to fit the width of the display field.</p> <p>NOTE: Ensure that the string displays properly in the UI—especially that the string does not have too many lines for the allocated display area.</p>
DoThumbnails Generate Thumbnails	boolean	false	<p>NOTE: This parameter must be present in the Adobe PDF settings file to take effect.</p> <p>If true, Distiller creates thumbnails for the pages of the resulting PDF file.</p>

TABLE 3.1 General parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
EndPage Page Ranges (All, From:, To:)	integer	-1	<p>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.</p> <p>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.</p>
HWResolution Resolution	array	[600 600]	<p>Provides the resolution for the PDF file if this value has not already been supplied by the PostScript file itself. Strictly speaking, this is not a Distiller parameter. See <i>PostScript Language Reference, third edition</i> for more information.</p>
ImageMemory	integer	1048576	<p>NOTE: This parameter does not appear in the UI.</p> <p>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.</p> <p>NOTE: If this parameter is set to a negative integer, Distiller defaults it to 0.</p>

TABLE 3.1 General parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
Optimize Optimize for fast web view	boolean	true	NOTE: This parameter must be present in the Adobe PDF settings file to take effect. If true , Distiller optimizes the PDF file. See the <i>PDF Reference</i> for more information on optimization (called <i>linearization</i> in that document).
PageSize Default Page Size	array	[612.000 792.000]	Provides the page size for the PDF file if this value has not already been supplied by the PostScript file itself. Strictly speaking, this is not a Distiller parameter. See <i>PostScript Language Reference, third edition</i> for more information.
StartPage Page Ranges (All, From:, To:)	integer	1	See the description of the EndPage parameter.

Image Compression Parameters

[Table 3.2](#), [Table 3.3](#), [Table 3.4](#), and [Table 3.5](#) list the Acrobat Distiller image 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 Images panel of the Distiller UI is shown just below the parameter name.

Color Image Compression Parameters

Table 3.2, “Color image compression parameters,” 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	“Factory” Default Value	Semantics
AntiAliasColorImages	boolean	false	<p>NOTE: This parameter does not appear in the UI.</p> <p>If true, Distiller permits anti-aliasing on color images. If false, Distiller does not permit anti-aliasing. 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 “Controlling Bit Depth” on page 32.</p>
AutoFilterColorImages Compression	boolean	true	<p>Distiller uses only if EncodeColorImages is true. If AutoFilterColorImages is true, Distiller inspects incoming images and selects JPEG compression for low-frequency images (images with smooth color changes) and lossless Flate compression for high-frequency images (images with sharp color changes). If false, Distiller compresses all color sampled images using the filter specified by ColorImageFilter. For a summary of Distiller’s use of image dictionaries based on how this parameter is set, see “Use of Image Dictionaries for Color and Grayscale Image Compression” on page 38.</p>

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
ColorACSIImageDict Compression Image Quality	dictionary	/QFactor 0.76 /HSamples [2 1 1 2] /VSamples [2 1 1 2]	Dictionary of parameters for JPEG compression when JPEG is chosen from the Automatic filter selection (see AutoFilterColorImages). ColorACSIImageDict is based on the DCTEncode parameter dictionary described in Section 3.13.3 in the <i>PostScript Language Reference, third edition</i> . See “ Setting Compression of Color and Grayscale Images ” on page 25 for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.
ColorImageAutoFilterStrategy Compression	name	JPEG	Must be one of the following values: JPEG If AutoFilterColorImages is set to true , Distiller uses lossy JPEG compression for low-frequency images and lossless Flate compression for high-frequency images. JPEG2000 If AutoFilterColorImages is set to true and if CompatibilityLevel is set to 1.5 or higher, then Distiller uses lossy JPEG2000 compression for low-frequency images (images with smooth color changes) and lossless JPEG2000 compression for high-frequency images. For a summary of Distiller’s decision flow for color and grayscale image compression, see “ Decision Flow for Color and Grayscale Image Compression ” on page 37.

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
ColorImageDepth	integer	-1	<p>NOTE: This parameter does not appear in the UI.</p> <p>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).</p>
ColorImageDict Compression Image Quality	dictionary	<code>/QFactor</code> <code>0.76</code> <code>/HSamples</code> <code>[2 1 1 2]</code> <code>/VSamples</code> <code>[2 1 1 2]</code>	<p>Dictionary of parameters for JPEG compression. ColorImageDict is based on the <code>DCTEncode</code> parameter dictionary described in Section 3.13.3 in the <i>PostScript Language Reference, third edition</i>.</p> <p>See “Setting Compression of Color and Grayscale Images” on page 25 for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.</p>
ColorImageDownsampleThreshold pixels-per-inch for images above: <i>value</i> pixels-per-inch	number	1.50000	<p>Sets the downsample threshold for color images. This is the ratio of image resolution to output resolution above which downsampling may be performed. Must be between 1.0 through 10.0, inclusive. If you set the threshold out of range, it reverts to a default of 1.5.</p> <p>See “Setting The Threshold Resolution For Downsampling Images” on page 29 for details on using this parameter.</p>

TABLE 3.2 *Color image compression parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
ColorImageDownsampleType Downsample: Average Downsampling to Subsampling to Bicubic Downsampling to Off	name	Bicubic	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. None

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
ColorImageFilter Compression: Off ZIP JPEG JPEG2000 Automatic(JPEG) Automatic(JPEG2000) NOTE: JPEG2000 options only appear in UI if CompatibilityLevel is set to 1.5 or higher.	name	DCTEncode	<p>Specifies the compression filter to be used for color images. Ignored if AutoFilterColorImages is true or EncodeColorImages is false. Valid names are:</p> <ul style="list-style-type: none"> DCTEncode Selects JPEG compression. FlateEncode Selects Flate (ZIP) compression. JPXEncode Selects JPEG2000 compression. <p>If DCTEncode is specified, it is only used if the output image has 8 bits-per-color component, that is,</p> <ul style="list-style-type: none"> if ColorImageDepth is 8 if ColorImageDepth is -1 and the original image has 8 bits-per-color component <p>Otherwise FlateEncode is used.</p> <p>For compatibility with Distiller 3.0 Adobe PDF settings files, Distiller 6.0 and later silently reverts to Flate compression if this parameter is set to LZWEncode. Distiller stops with a range error if this parameter is set to one of these:</p> <ul style="list-style-type: none"> CCITTFaxEncode RunLengthEncode ASCIIHEXEncode ASCII85Encode InvalidEncodeXYZ

TABLE 3.2 *Color image compression parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
ColorImageMinDownsampleDepth	integer	1	<p>NOTE: This parameter does not appear in the UI.</p> <p>If DownsampleColorImages is true, controls the range of bit depths for which color image downsampling occurs. Valid values are 1, 2, 4 or 8. For more information, see “Controlling the Range of Bit Depths For Which Downsampling Occurs” on page 29.</p>
ColorImageMinResolution Policy	integer	150	<p>Imposes a lower limit to the resolution of sampled images. The legal values are from 9 to 64000, inclusive. How this value is used by Distiller is determined by ColorImageMinResolutionPolicy. For more information, see “Specifying a Lower Limit for the Resolution of Sampled Images” on page 30.</p>

TABLE 3.2 *Color image compression parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
ColorImageMinResolutionPolicy Policy	name	OK	<p>Sets the policy for imposition of a lower limit to the resolution of sampled color images as specified by the ColorImageMinResolution parameter. Valid names are:</p> <p>OK</p> <p>Distiller's default behavior does not change—i.e., Distiller does not enforce any lower limit on image resolution, ignoring any value specified by ColorImageMinResolution.</p> <p>Warning</p> <p>A warning is issued every time a sampled color image with resolution smaller than the value specified by ColorImageMinResolution is placed in the PDF file. The job continues after issuing the warning.</p> <p>Error</p> <p>An error occurs when a sampled color image with resolution smaller than the value specified by ColorImageMinResolution is placed in the PDF file. The job fails with a limit check error.</p> <p>For more information, see “Specifying a Lower Limit for the Resolution of Sampled Images” on page 30.</p>

TABLE 3.2 Color image compression parameters

Parameter/ UI Name	Type	"Factory" Default Value	Semantics
ColorImageResolution pixels-per-inch	integer	150	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" on page 29 . Distiller downsamples to this minimum resolution. The legal values are from 9 to 2400, inclusive.
ConvertImagesToIndexed	boolean	true	NOTE: This parameter does not appear in the UI. 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.
CropColorImages	boolean	true	NOTE: This parameter does not appear in the UI. If CropColorImages is false , then color images will never be cropped, whether or not the current clip would remove any image samples. Color images are images with more than one color component. For more information, see "Disabling of Image Cropping" on page 41 .
DownsampleColorImages Downsample	boolean	true	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	"Factory" Default Value	Semantics
EncodeColorImages Compression	boolean	true	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.
JPEG2000ColorACSIImageDict Compression Image Quality Tile Size NOTE: Tile size can only be set from the UI if CompatibilityLevel is set to 1.5 or higher, and the Compression setting is Automatic (JPEG2000).	dictionary	/TileWidth 256 /TileHeight 256 /Quality 15	Dictionary of parameters for automatic JPEG2000 compression. This dictionary cannot be used at the PostScript language level in a PostScript file that is being processed by Distiller. See "JPEG2000 Compression" on page 21 for details.
JPEG2000ColorImageDict Compression Image Quality Tile Size NOTE: Tile size can only be set from the UI if CompatibilityLevel is set to 1.5 or higher, and the Compression setting is Automatic (JPEG2000).	dictionary	/TileWidth 256 /TileHeight 256 /Quality 15	Dictionary of parameters for JPEG2000 compression. This dictionary cannot be used at the PostScript language level in a PostScript file that is being processed by Distiller. See "JPEG2000 Compression" on page 21 for details.

Grayscale Image Compression Parameters

Table 3.3, "Grayscale image compression parameters," 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	“Factory” Default Value	Semantics
AntiAliasGrayImages	boolean	false	<p>NOTE: This parameter does not appear in the UI.</p> <p>If true, Distiller permits anti-aliasing on grayscale images. If false, Distiller does not permit 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. 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 “Controlling Bit Depth” on page 32.</p>
AutoFilterGrayImages Compression	boolean	true	<p>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, for example, 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. For a summary of Distiller’s use of image dictionaries based on how this parameter is set, see “Use of Image Dictionaries for Color and Grayscale Image Compression” on page 38.</p>

TABLE 3.3 *Grayscale image compression parameters*

Parameter/ UI Name	Type	"Factory" Default Value	Semantics
CropGrayImages	boolean	true	<p>NOTE: This parameter does not appear in the UI.</p> <p>If CropGrayImages is false, then gray images will never be cropped, whether or not the current clip would remove any image samples. Gray images are images with only one color component, and more than one bit per sample. For more information, see "Disabling of Image Cropping" on page 41.</p>
DownsampleGrayImages Downsample	boolean	true	<p>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.</p>
EncodeGrayImages Compression	boolean	true	<p>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.</p>

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
GrayACSIImageDict Compression Image Quality	dictionary	/QFactor 0.76 /HSamples [2 1 1 2] /VSamples [2 1 1 2]	Dictionary of parameters for JPEG compression when JPEG is chosen from the Automatic filter selection (see AutoFilterGrayImages). GrayACSIImageDict is based on the DCTEncode parameter dictionary described in Section 3.13.3 in the <i>PostScript Language Reference, third edition</i> . See “Setting Compression of Color and Grayscale Images” on page 25 for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.
GrayImageAutoFilterStrategy Compression	name	JPEG	Must be one of the following values: JPEG If AutoFilterColorImages is set to true , Distiller uses lossy JPEG compression for low-frequency images and lossless Flate compression for high-frequency images. JPEG2000 If AutoFilterColorImages is set to true and if CompatibilityLevel is set to 1.5 or higher, then Distiller uses lossy JPEG2000 compression for low-frequency images (images with smooth color changes) and lossless JPEG2000 compression for high-frequency images. For a summary of Distiller’s decision flow for color and grayscale image compression, see “Decision Flow for Color and Grayscale Image Compression” on page 37 .

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
GrayImageDepth	integer	-1	<p>NOTE: This parameter does not appear in the UI.</p> <p>Specifies the number of bits-per-sample in the downsampled image when DownsampleGrayImages is true.</p> <p>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).</p>
GrayImageDict Compression Image Quality	dictionary	/QFactor 0.76 /HSamples [2 1 1 2] /VSamples [2 1 1 2]	<p>Dictionary of parameters for JPEG compression. GrayImageDict is based on the DCTEncode parameter dictionary described in Section 3.13.3 in the <i>PostScript Language Reference, third edition</i>. See “Setting Compression of Color and Grayscale Images” on page 25 for details on the keys Distiller sets in this dictionary and how to change settings to customize compression.</p>
GrayImageDownsampleThreshold pixels-per-inch for images above: <i>value</i> pixels-per-inch	number	1.50000	<p>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” on page 29 for details on using this parameter.</p>

TABLE 3.3 *Grayscale image compression parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
GrayImageDownsampleType Downsample: Average Downsampling to Subsampling to Bicubic Downsampling to Off	name	Bicubic 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. None	Must be one of the following values:

TABLE 3.3 Grayscale image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
GrayImageFilter Compression: Off ZIP JPEG JPEG2000 Automatic(JPEG) Automatic(JPEG2000) NOTE: JPEG2000 options only appear in UI if CompatibilityLevel is set to 1.5 or higher.	name	DCTEncode	<p>Specifies the compression filter to be used for grayscale images. Distiller ignores if AutoFilterGrayImages is true or EncodeGrayImages is false. Valid names are:</p> <ul style="list-style-type: none"> DCTEncode Selects JPEG compression. FlateEncode Selects Flate compression. JPXEncode Selects JPEG2000 compression. <p>If DCTEncode is specified, it is only used if the output image has 8 bits-per-sample, that is,</p> <ul style="list-style-type: none"> 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 MonolImageDepth is 8 <p>Otherwise Distiller uses FlateEncode.</p> <p>For compatibility with Distiller 3.0 Adobe PDF settings files, Distiller 6.0 and later silently reverts to Flate compression if this parameter is set to LZWEncode. Distiller stops with a range error if this parameter is set to one of these:</p> <ul style="list-style-type: none"> CCITTFaxEncode RunLengthEncode ASCIIHEXEncode ASCII85Encode InvalidEncodeXYZ

TABLE 3.3 *Grayscale image compression parameters*

Parameter/ UI Name	Type	"Factory" Default Value	Semantics
GrayImageMinDownsampleDepth	integer	2	<p>NOTE: This parameter does not appear in the UI.</p> <p>If DownsampleGrayImages is true, controls the range of bit depths for which gray image downsampling occurs. Valid values are 2, 4 or 8. For more information, see "Controlling the Range of Bit Depths For Which Downsampling Occurs" on page 29.</p>
GrayImageMinResolution Policy	integer	150	<p>Imposes a lower limit to the resolution of sampled grayscale images. The legal values are from 9 to 64000, inclusive. How this value is used by Distiller is determined by GrayImageMinResolutionPolicy. For more information, see "Specifying a Lower Limit for the Resolution of Sampled Images" on page 30.</p>

TABLE 3.3 *Grayscale image compression parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
GrayImageMinResolutionPolicy Policy	name	OK	<p>Sets the policy for imposition of a lower limit to the resolution of sampled images as specified by the GrayImageMinResolution parameter. Valid names are:</p> <p>OK</p> <p>Distiller's default behavior does not change—i.e., Distiller does not enforce any lower limit on image resolution, ignoring any value specified by GrayImageMinResolution.</p> <p>Warning</p> <p>A warning is issued every time a sampled grayscale image with resolution smaller than the value specified by GrayImageMinResolution is placed in the PDF file. The job continues after issuing the warning.</p> <p>Error</p> <p>An error occurs when a sampled grayscale image with resolution smaller than the value specified by GrayImageMinResolution is placed in the PDF file. The job fails with a limit check error.</p> <p>For more information, see “Specifying a Lower Limit for the Resolution of Sampled Images” on page 30.</p>

TABLE 3.3 *Grayscale image compression parameters*

Parameter/ UI Name	Type	"Factory" Default Value	Semantics
GrayImageResolution pixels-per-inch	integer	150	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" on page 29 . Distiller downsamples to this minimum resolution. The legal values are from 9 to 2400, inclusive.
JPEG2000GrayACSIImageDict Compression Image Quality Tile Size NOTE: Tile size can only be set from the UI if CompatibilityLevel is set to 1.5 or higher, and the Compression setting is Automatic (JPEG2000).	dictionary	/TileWidth 256 /TileHeight 256 /Quality 15	Dictionary of parameters for automatic JPEG2000 compression. This dictionary cannot be used at the PostScript language level in a PostScript file that is being processed by Distiller. See "JPEG2000 Compression" on page 21 for details.
JPEG2000GrayImageDict Compression Image Quality Tile Size NOTE: Tile size can only be set from the UI if CompatibilityLevel is set to 1.5 or higher, and the Compression setting is Automatic (JPEG2000).	dictionary	/TileWidth 256 /TileHeight 256 /Quality 15	Dictionary of parameters for JPEG2000 compression. This dictionary cannot be used at the PostScript language level in a PostScript file that is being processed by Distiller. See "JPEG2000 Compression" on page 21 for details.

Monochrome Images Compression Parameters

Table 3.4, “Monochrome image compression parameters,” lists the monochrome image compression parameters. These parameters control the compression and downsampling of

monochrome images. See [“Setting Compression of Monochrome \(Black and White\) Images” on page 27](#) for details on using these parameters.

NOTE: With the exception of the **AntiAliasMonolImages** and **MonolImageDepth** 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	“Factory” Default Value	Semantics
AntiAliasMonolImages Anti-alias to gray	boolean	false	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 MonolImageDepth has a value greater than 1. For more information on anti-aliasing see “Controlling Bit Depth” on page 32 . NOTE: Distiller does not do anti-aliasing for image masks, regardless of the value of AntiAliasMonolImages .
CropMonolImages	boolean	true	NOTE: This parameter does not appear in the UI. If CropMonolImages is false , then monochrome images will never be cropped, whether or not the current clip would remove any image samples. Monochrome images are images with only one color component, and only one bit per sample. For more information, see “Disabling of Image Cropping” on page 41 .

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
DownsampleMonolImages Downsample	boolean	true	If true , Distiller downsamples monochrome images using the resolution specified by MonolImageResolution . 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.
EncodeMonolImages Compression	boolean	true	If true , Distiller encodes monochrome images using the compression filter specified by the value of the MonolImageFilter key. If false , Distiller does not apply any compression filters to monochrome images.
MonolImageDepth Anti-alias to gray	integer	-1	Specifies the number of bits-per-sample in the downsampled image when DownsampleMonolImages 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 MonolImageDepth is 2 , 4 , or 8 , monochrome images are converted to grayscale images. Distiller does not use MonolImageDepth unless AntiAliasMonolImages is true. NOTE: Distiller ignores MonolImageDepth for image masks.
MonolImageDict Compression Quality	dictionary	/K -1	Dictionary of parameters for CCITT FaxEncode compression. MonolImageDict is based on the CCITT FaxEncode parameter dictionary. See “ Setting Compression of Monochrome (Black and White) Images ” on page 27 for details on Distiller’s settings and customizing CCITT FaxEncode compression.

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
MonolImageDownsampleThreshold pixels-per-inch for images above: <i>value</i> pixels-per-inch	number	1.50000	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 ” on page 29 for details on using this parameter.
MonolImageDownsampleType Downsample: Average Downsampling to Subsampling to Bicubic Downsampling to Off	name	Bicubic Average Bicubic Subsample None	Must be one of the following values: Distiller averages groups of samples to get the new downsampled value. Distiller uses bicubic interpolation on a group of samples to get a new downsampled value. Distiller picks the middle sample from a group of samples to get the new downsampled value.

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
MonolImageFilter Compression: CCITT Group 3 CCITT Group 4 Run Length ZIP Off	name	CCITTFa xEncode	<p>Specifies the compression filter Distiller will use for monochrome images. Valid names are:</p> <p>CCITTFaxEncode Selects CCITT Group 3 or 4 facsimile encoding.</p> <p>FlateEncode Selects Flate compression.</p> <p>RunLengthEncode Selects run length encoding.</p> <p>For compatibility with Distiller 3.0 Adobe PDF settings files, Distiller 6.0 and later silently reverts to Flate compression if this parameter is set to LZWEncode. Distiller stops with a range error if this parameter is set to one of these:</p> <ul style="list-style-type: none"> • DCTEncode • ASCIIHEXEncode • ASCII85Encode • JPXEncode • InvalidEncodeXYZ
MonolImageMinResolution Policy	integer	300	Imposes a lower limit to the resolution of sampled monochrome images. The legal values are from 9 to 64000 , inclusive. How this value is used by Distiller is determined by MonolImageMinResolutionPolicy . For more information, see “ Specifying a Lower Limit for the Resolution of Sampled Images ” on page 30.

TABLE 3.4 Monochrome image compression parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
MonolImageMinResolutionPolicy Policy	name	OK	<p>Sets the policy for imposition of a lower limit to the resolution of sampled images as specified by the MonolImageMinResolution parameter. Valid names are:</p> <p>OK Distiller's default behavior does not change—i.e., Distiller does not enforce any lower limit on image resolution, ignoring any value specified by MonolImageMinResolution.</p> <p>Warning A warning is issued every time a sampled monochrome image with resolution smaller than the value specified by MonolImageMinResolution is placed in the PDF file. The job continues after issuing the warning.</p> <p>Error An error occurs when a sampled monochrome image with resolution smaller than the value specified by MonolImageMinResolution is placed in the PDF file. The job fails with a limit check error.</p> <p>For more information, see “Specifying a Lower Limit for the Resolution of Sampled Images” on page 30.</p>
MonolImageResolution pixels-per-inch	integer	300	Specifies the minimum resolution for downsampled monochrome images. This value is used only when DownsampleMonolImages is true and the resolution of the input image meets the criteria described in “ Setting The Threshold Resolution For Downsampling Images ” on page 29. Distiller downsamples to this minimum resolution. The legal values are from 9 to 2400, inclusive.

Page Compression Parameter

[Table 3.5, “Page compression parameter,”](#) describes the page compression parameter.

TABLE 3.5 *Page compression parameter*

Parameters	Type	“Factory” Default Value	Semantics
CompressPages	boolean	true	<p>NOTE: This parameter does not appear in the UI.</p> <p>If true, Distiller uses Flate compression to compress page content streams as well as form, pattern, and Type 3 font content streams.</p>

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, “Font embedding and subsetting parameters,”](#) lists the parameters available for controlling font embedding and subsetting. The corresponding name of each parameter in

the Fonts panel of the Distiller UI is shown just below the parameter name. For more information on font embedding, see ["Using the Font Parameters" on page 41](#).

TABLE 3.6 *Font embedding and subsetting parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
AlwaysEmbed Always Embed	array	[true]	<p>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:</p> <p>If the array consists entirely of names:</p> <ul style="list-style-type: none"> • Distiller sets its internal list of fonts that must be embedded to be exactly the list of names in the array. <p>If the first array value is a boolean:</p> <ul style="list-style-type: none"> • 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. <p>See EmbedAllFonts for an explanation of how the Distiller resolves ambiguity in the use of this parameter, EmbedAllFonts, and NeverEmbed.</p>
CannotEmbedFontPolicy When embedding fails	name	Warning	<p>The policy Distiller uses if it cannot find, or cannot embed, the font. The possible name values are:</p> <p>OK Distiller ignores and continues.</p> <p>Warning Distiller displays a warning and continues.</p> <p>Error Distiller quits distilling the current job.</p>

TABLE 3.6 **Font embedding and subsetting parameters**

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
EmbedAllFonts Embed all fonts	boolean	true	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.
EmbedOpenType Embed OpenType fonts Note: Embed OpenType fonts can only be set from the UI if CompatibilityLevel is set to 1.6 or higher.	boolean	false	If true , OpenType fonts are embedded if: <ul style="list-style-type: none"> • The OpenType font is to be embedded within a Type 1 font descriptor, • EmbedOpenType is true, • CompatibilityLevel is 1.6 (or higher), and • SubsetFonts is false, or SubsetFonts is true and MaxSubsetPct n and “percent of characters used” is greater than or equal to n.
MaxSubsetPct Subset embedded fonts when percent of characters used is less than: <i>value</i> %	integer	100	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 . For example, a value of 30 means that a font will be embedded in full (not subset) if more than 30% of glyphs are used; a value of 100 means all fonts will be subset no matter how many glyphs are used (because you cannot use more than 100% of glyphs).

TABLE 3.6 **Font embedding and subsetting parameters**

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
NeverEmbed Never Embed	array	[true]	<p>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:</p> <p>If the array consists entirely of names:</p> <ul style="list-style-type: none"> • Distiller sets its internal list of fonts that must never be embedded to be exactly the list of names in the array. <p>If the first array value is a boolean:</p> <ul style="list-style-type: none"> • If the value is true, Distiller adds the font names in the rest of the NeverEmbed 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. <p>See EmbedAllFonts in this section for an explanation of how the Distiller resolves ambiguity in the use of this parameter, EmbedAllFonts, and NeverEmbed.</p> <p>When creating the array of font names, you cannot add the font names to both the AlwaysEmbed and NeverEmbed lists.</p> <p>See EmbedAllFonts for an explanation of how the Distiller resolves ambiguity in the use of this parameter, EmbedAllFonts, and NeverEmbed.</p>

TABLE 3.6 *Font embedding and subsetting parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
SubsetFonts Subset embedded fonts when percent of characters used is less than:	boolean	true	<p>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.</p> <p>Subsetted fonts in the PDF file appear with a 6-letter prefix and a plus (+) sign. For example, Palatino subsetted may appear as:</p> <p style="text-align: center;">NPBOME+Palatino-Roman</p> <p>NOTE: Embedded instances of multiple master fonts and of Type 3, TrueType, and CID fonts are always subsetted, regardless of the value of SubsetFonts.</p>

Color Conversion Parameters

[Table 3.7, “Color conversion parameters,”](#) lists the parameters color conversion parameters. The corresponding name, if any, of each parameter in the Color panel of the Distiller UI is shown just below the parameter name. See “ICCBased Color Spaces” in Section 4.5.4 of the *PDF Reference*, for details on profiles.

TABLE 3.7 *Color conversion parameters*

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
CalRGBProfile Working Spaces: RGB	string	(sRGB IEC61966 -2.1)	The name of the ICC profile that is used for tagging or converting RGB images, text, and/or graphics.

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
CalCMYKProfile Working Spaces: CMYK	string	(U.S. Web Coated SWOP v2)	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.
ColorConversionStrategy Color Management Policies	name	sRGB	Sets the ICC profile used for color conversion strategy. See “ ColorConversionStrategy Parameter Details ” on page 44 for details on how to use this parameter. Legal values of this parameter are: LeaveColorUnchanged UseDeviceIndependentColor UseDeviceIndependentColor-ForImages sRGB CMYK More information on this parameter can also be found in the <i>Acrobat Guide</i> in Distiller online Help.
ColorSettingsFile Settings File	string	()	See the <i>Acrobat Guide</i> in Distiller online Help for information on color management policies and how to use this parameter.

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	Type	"Factory" Default Value	Semantics
DefaultRenderingIntent Intent	name	Default	<p>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 <i>PostScript Language Reference, second edition</i>. 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, no rendering intent is written to the PDF. Legal values of this parameter are:</p> <p>Default Perceptual Saturation RelativeColorimetric AbsoluteColorimetric</p>
ParseICCProfilesInComments	boolean	true	<p>NOTE: This parameter does not appear in the UI.</p> <p>If true, Distiller honors EPS embedded ICC profiles when distilling. ICC profiles are honored only if they are enclosed in two DSC pairs: ICCProfile and SetColorSpace. See the ICC specification (available at http://www.color.org), section B.2, for details on the syntax of these comment pairs.</p> <p>NOTE: This parameter is ignored if CompatibilityLevel is set to 1.2.</p>

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
PreserveDICMYKValues Preserve CMYK values for calibrated CMYK color spaces	boolean	true	Describes what to do with color values for Device Independent CMYK color spaces. This parameter is used only if ColorConversionStrategy is CMYK . If true , CIEBasedDEFG CMYK color values are treated as DeviceCMYK values; CIEBasedDEFG color spaces will be ignored and discarded. If false , a conversion from CIEBasedDEFG color space to CMYK working space is performed.
PreserveHalftoneInfo Preserve Halftone Information	boolean	false	If true , Distiller passes halftone screen information (frequency, angle, and spot function) into the PDF file. If false , halftone information is not passed in.
sRGBProfile	string	(sRGB IEC61966 -2.1)	NOTE: This parameter does not appear in the UI. <i>(Read Only)</i> 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).

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
TransferFunctionInfo When transfer functions are found	name	Preserve	<p>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:</p> <p>Preserve Distiller preserves (passes into the PDF file) transfer functions.</p> <p>Remove Distiller ignores transfer functions. They are neither applied to the color values by Distiller nor passed into the PDF file.</p> <p>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 and later. It is sometimes used to achieve artistic effects (although the <i>PostScript Language Reference</i> discourages such usage).</p> <p>NOTE: If you are generating PDF/X-compliant files, do not set this to Preserve.</p>

TABLE 3.7 Color conversion parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
UCRandBGInfo Preserve under color removal and black generation	name	Remove	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 <i>PostScript Language Reference, third edition</i> , for details on the setundercolorremoval and setblackgeneration operators and descriptions undercolor removal (UCR) and black generation (BG).

Advanced Adobe PDF Settings

Table 3.8, “Advanced parameters,” list the advanced Distiller parameters that you can customize. The corresponding name, if any, of each parameter in the Advanced panel of the Distiller UI is shown just below the parameter name.

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
AllowPSXObjects Allow PostScript XObjects	boolean	true	Allow PostScript XObjects. For a description of PostScript XObjects, see Section 4.9 of the <i>PDF Reference</i> .

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
AllowTransparency	boolean	false	<p>NOTE: This parameter does not appear in the UI.</p> <p>Used to control whether the <code>/SetTransparency</code> pdfmark is honored or rejected. When set to <code>true</code>, then <code>[... /SetTransparency</code> pdfmark is allowed in PS jobs if <code>CompatibilityLevel</code> is 1.4 or higher. If <code>AllowTransparency</code> is <code>false</code>, then <code>/SetTransparency</code> pdfmark is treated as error.</p> <p><code>/SetTransparency</code> pdfmark is a new pdfmark extension used to produce transparency in PDF. For more details, see the <i>pdfmark Reference Manual</i>.</p> <p>NOTE: This parameter is used when generating PDF 1.4 or later.</p>
ASCII85EncodePages	boolean	false	<p>NOTE: This parameter does not appear in the UI.</p> <p>If <code>true</code>, 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 <code>false</code>, 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.</p>
AutoPositionEPSFiles Resize page and center artwork for EPS files	boolean	true	<p>If <code>true</code>, Distiller resizes the created page to the size of the EPS file using the <code>%%BoundingBox</code> 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 <code>ParseDSCComments</code> is <code>false</code>.</p>

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
CreateJDFFile Create Job Definition Format (JDF) File NOTE: This parameter appears in the Acrobat Professional UI for versions 6.0 and above, though the parameter appears in all settings files regardless of model.	boolean	false	If true , a JDF file is created. If false , a JDF file is not created. See “CreateJDFFile Parameter Details” on page 49 for more information.
CreateJobTicket Save Portable Job Ticket inside PDF file	boolean	false	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 <i>Portable Job Ticket Format, version 1.1</i> (Technical Note #5620). See “CreateJobTicket Parameter Details” on page 47 for details on this parameter.

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
DetectCurves Convert smooth lines to curves	number	0 . 1	<p>If 0 . 0000, then this feature is disabled (Off). If 0 . 1000, then this feature is enabled (On). Value must be in the range from 0 . 0000 to 10 . 0000. Positive values cause Distiller to investigate graphics for curves that are not described efficiently and that thus result in unacceptably large file sizes. For these curves, Distiller provides a conversion into correct Bezier curves that look the same but take up much less file space. Visual inspection of the results suggests that the 0 . 1000 value yields the closest approximation to the original curve. The value is used in user space (72 dpi) to control the accuracy of Distiller’s curve-fitting algorithm. The curve-fitting results should not part from the original line segments more than this number. Distiller generates a range error if the value is less than 0 . 0 or greater than 10 . 0.</p>
DSCReportingLevel	integer	0	<p>NOTE: This parameter does not appear in the UI.</p> <p>Level can be either 0, 1, or 2. 0 means no additional reporting. Level 1 shows all input as it is parsed and shows a tree crawl when getting into bad states. Level 2 shows transitions in addition to the information in Level 1.</p>

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
DetectBlends	boolean	true	<p>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.</p> <p>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.</p> <p>See “Converting Gradients to Smooth Shades” on page 46 for a discussion of DetectBlends and idiom recognition.</p>
EmbedJobOptions	boolean	false	<p>If true, the Adobe PDF settings file used to create the PDF is embedded in the PDF and is accessible through Document > File Attachments from the Acrobat UI.</p> <p>NOTE: Inside the PDF file, the Adobe PDF settings file becomes an item in the Names->EmbeddedFiles tree (see <i>PDF Reference</i>, Section 3.6.3).</p>
EmitDSCWarnings	boolean	false	<p>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.</p>
LockDistillerParams	boolean	false	<p>If true, Distiller ignores any parameters defined in the incoming PostScript file (in this case, the Distiller parameters are defined by the Adobe PDF settings as set in the UI). If false, the Distiller parameters, if any, defined in the PostScript file override those defined by the Adobe PDF settings set in the UI.</p>

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
OPM Overprinting default is nonzero overprinting	integer	1	<p>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, <i>Color Separation Conventions for PostScript Language Programs</i>, and Section 4.5.6 of the <i>PDF Reference</i>.</p> <p>NOTE: Distiller ignores this parameter if PreserveOverprintSettings is false.</p>
ParseDSCComments Process DSC Comments	boolean	true	<p>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.</p>
ParseDSCCommentsForDocInfo Preserve document information from DSC	boolean	true	<p>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 Properties windows of the UI. Distiller 5.0 also embeds the Document Information as XML in the PDF file. See “ParseDSCCommentsForDocInfo Parameter Details” on page 46 for details on this parameter.</p> <p>NOTE: If true, document properties of Microsoft Office files are carried into the PDF. Setting this parameter to false prevents this transfer of information.</p>
PassThroughJPEGImages Save original JPEG images in PDF if possible	boolean	false	<p>Distiller ignores this parameter if ParseDSCComments is false.</p> <p>If true, Distiller passes JPEG images through without re-compressing them. The advantages and disadvantages to doing this are described in “Passing Through JPEG Images” on page 39.</p>

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
PreserveCopyPage Preserve Level 2 copypage semantics	boolean	true	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 <i>PostScript Language Reference, third edition</i> , for more information.
PreserveEPSInfo Preserve EPS information from DSC	boolean	false	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 (and above) also embeds the EPS information as XML in a Metadata dictionary attached to the Marked Content. See “ PreserveEPSInfo Parameter Details ” on page 47 for details on this parameter. Distiller ignores this parameter if ParseDSCComments is false .
PreserveFlatness	boolean	true	NOTE: This parameter does not appear in the UI. If true , the PostScript flatness set by the setflat operator will be preserved. If false , flatness will be discarded. Preserving flatness can increase rendering and printing speeds, since less time is spent determining how to precisely render curves and circles.

TABLE 3.8 Advanced parameters

Parameter/ UI Name	Type	“Factory” Default Value	Semantics
PreserveOPIComments Preserve OPI comments	boolean	false	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
PreserveOverprintSettings Preserve overprint settings	boolean	true	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).
UsePrologue Use Prologue.ps and Epilogue.ps	boolean	false	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.

Standards Parameters (Acrobat Professional Only)

Table 3.9, “Standards parameters,” lists the PDF/X Compliance Check and PDF/X Enhancement parameters. The corresponding name of each parameter in the PDF/X panel

of the Distiller UI is shown just below the parameter name. See ["Using the Standards Parameters" on page 50](#) for more information.

TABLE 3.9 **Standards parameters**

Parameters			"Factory" Default Value	Semantics
UI Name	Type			
CheckCompliance Compliance Standard	array	[/None]	Determines the standard whose compliance will be checked for the current job, if any. Valid values of this parameter are: None PDFA:DRAFT PDFX1a:2001 PDFX1a:2003 PDFX3:2002 PDFX3:2003	<p>NOTE: In Distiller 7, this parameter takes precedence over the PDFX1aCheck and PDFX3Check parameters. If CheckCompliance is used, the values of PDFX1aCheck and/or PDFX3Check are ignored.</p> <p>For more information on how this parameter works, see "Using the Standards Parameters" on page 50.</p>
PDFX1aCheck	boolean	false	<p>NOTE: This parameter does not appear in the UI.</p> <p>If true, checks compliance with the PDF/X-1a standard (ISO 15930-1:2001) and a PDF/X compliance report is written to the message log. A value of /PDFX1aCheck true is equivalent in Distiller 7 to a value of /CheckCompliance [/PDFX1a:2001].</p> <p>NOTE: This parameter is retained for compatibility with Distiller 6. For more information on how this parameter works, see "Using the Standards Parameters" on page 50.</p>	

TABLE 3.9 Standards parameters

Parameters		Type	“Factory” Default Value	Semantics
UI Name				
PDFX3Check	boolean	false		<p>NOTE: This parameter does not appear in the UI.</p> <p>If true, checks compliance with the PDF/X-3 standard (ISO 15930-3:2002) and a PDF/X compliance report is written to the message log. A value of /PDFX3Check true is equivalent in Distiller 7 to a value of /CheckCompliance [/PDFX3:2002].</p>
				<p>NOTE: This parameter is retained for compatibility with Distiller 6. For more information on how this parameter works, see “Using the Standards Parameters” on page 50.</p>
PDFXCompliantPDFOnly When not compliant: Continue Cancel job	boolean	false		<p>If true, Distiller produces a PDF document only if PDF/X compliance tests are passed. If false, Distiller performs validation checks, but does not insert PDF/X additional key/value pairs into the created PDF file, nor does Distiller fail if validation errors are revealed.</p>
PDFXNoTrimBoxError Report as error	boolean	true		<p>If true and both the TrimBox and ArtBox entries are not specified in the page object, the condition is reported as an error.</p>
PDFXTrimBoxtoMediaBoxOffset Set TrimBox to MediaBox with offsets (units) This parameter is ignored if PDFXNoTrimBoxError is true .	array	[0.00000 0.00000 0.00000 0.00000]		<p>If both the TrimBox and ArtBox entries are not specified in the page object, TrimBox is set to MediaBox with offsets. Offsets are specified as [left right top bottom]. All numbers must be greater than or equal to 0.0. TrimBox offsets place the TrimBox entirely inside the MediaBox.</p>
PDFXSetBleedBoxToMediaBox Set BleedBox to MediaBox	boolean	true		<p>If true and the BleedBox entry is not specified in the page object, BleedBox is set to MediaBox.</p>

TABLE 3.9 Standards parameters

Parameters			“Factory” Default Value	Semantics
UI Name	Type			
PDFXBleedBoxToTrimBoxOffset Set BleedBox to TrimBox with offsets (units)	array	[0.00000 0.00000 0.00000 0.00000]		If the BleedBox entry is not specified in the page object, BleedBox is set to TrimBox with offsets. Offsets are specified as [left right top bottom] . All numbers must be greater than or equal to 0.0. BleedBox offsets place the BleedBox entirely outside the TrimBox .
NOTE: This parameter is ignored if PDFXSetBleedBoxToMediaBox is true.				
PDFXOutputConditionIdentifier Output Condition Identifier	string	()		The PDF/X Output Condition Identifier is a Reference Name that is specified by the Output Intent Profile Name’s registry. The entry is automatically entered for known Output Intent Profile Names. If Use Output Condition Identifier is specified for PDFXOutputIntentProfile , then this parameter must be provided for PDF/X validation to succeed.
PDFXOutputIntentProfile Output Intent Profile Name	string	()		The Output Intent Profile Name indicates the characterized printing condition for which the document has been prepared and is required for PDF/X compliance. One of the provided names may be used, or a different name may be provided by specifying Use Output Condition Identifier (see PDFXOutputConditionIdentifier). (None) should be used for workflows that require that the document specify an output intent and for which compliance checking should fail if it is not present in the document. (None) displays as the localized menu string No Default Profile . NOTE: Values are string literals, so parentheses are required around them.
				For a description of how values are used to fill out entries in the PDF/X output intent dictionary, see “ Using the Standards Parameters ” on page 50.

TABLE 3.9 Standards parameters

Parameters	Type	“Factory” Default Value	Semantics
UI Name			
PDFXOutputCondition Output Condition	string	()	The PDF/X OutputCondition is an optional comment which, if present, is added to the PDF file and describes the intended printing condition in a form that should be meaningful to a human operator at the site receiving the PDF document.
PDFXRegistryName Registry Name (URL)	string	(http://www.color.org)	The PDF/X RegistryName is an optional string which, if present, preferably has a value that provides a URL at which more information regarding the registry may be obtained. This entry is automatically populated for recognized ICC profile names.
PDFXTrapped Trapped: Leave undefined Insert False Insert True	name	False	Indicates the state of trapping within the file. A value of True or False (as a name object, not the similarly-sounding boolean values true and false) is required for PDF/X compliance. If a document does not specify a Trapped state, then the value provided here is used. Unknown should be used for workflows that require that the document specify a Trapped state and for which compliance checking should fail if it is not present in the document. Can be one of the following values: Unknown False True

This chapter describes the changes made to this document for the last three major releases of Acrobat.

Changes Made for Acrobat 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 [“Setting Compression of Color and Grayscale Images” on page 25](#) and in [“Setting Compression of Monochrome \(Black and White\) Images” on page 27.](#)
- Updated the image compression settings in [Table 2.6](#).
- Added the Acrobat 5.0 UI names associated with the Distiller parameter names in [Chapter 3, “Parameter Reference.”](#)
- Updated the document for consistency and for technical accuracy.

Changes Made for Acrobat 6

- Added more detail to the description of how Distiller selects image compression filters.
- Added information relating to the new image compression filter for JPEG2000 encoding.
- Added summary of Distiller image compression controls.
- Added pseudocode showing how automatic compression works; added pseudocode showing how Distiller uses image dictionaries.
- Removed the **UseFlateCompression** parameter, which was deprecated in Distiller 4.0.
- Added “Factory Default Values” column to parameter description tables.
- Swapped out non-Adobe fonts from document.
- Added descriptions for six new predefined Adobe PDF settings files:
 - **Standard**
 - **High Quality**

- **PDFX1a**
- **PDFX3**
- **Smallest File Size**
- **Press Quality**
- Added new parameters information:
 - **JPEG2000** value for the **ColorImageFilter** parameter
 - **JPEG2000ColorImageDict** parameter
 - **JPEG2000ColorACSIImageDict** parameter
 - **JPEG2000** value for the **GrayImageFilter**
 - **JPEG2000GrayImageDict** parameter
 - **JPEG2000GrayACSIImageDict** parameter
 - **ColorImageAutoFilterStrategy** parameter
 - **GrayImageAutoFilterStrategy** parameter
 - **PassThroughJPEGImages** parameter
 - **AllowPSXObjects** parameter
 - **DSCReportingLevel** parameter
 - **PDFX1aCheck** parameter
 - **PDFX3Check** parameter
 - **PDFXCompliantPDFOnly** parameter
 - **PDFXNoTrimBoxError** parameter
 - **PDFXTrimBoxToMediaBoxOffset** parameter
 - **PDFXSetBleedBoxToMediaBox** parameter
 - **PDFXBleedBoxToTrimBoxOffset** parameter
 - **PDFXOutputIntentProfile** parameter
 - **PDFXTrapped** parameter
 - **PDFXOutputCondition** parameter
 - **PDFXRegistryName** parameter
 - **AllowTransparency** parameter
 - **CreateJDFFile** parameter
 - **EmbedJobOptions** parameter
 - **Description** parameter
 - **CompressObjects** parameter

Changes Made for Acrobat 7

- Added descriptions for three new predefined Adobe PDF settings files and noted the renaming of three others:
 - **High Quality Print** (was **High Quality**)
 - **PDFX1a 2001** (was **PDFX1a**)
 - **PDFX1a 2003**

- **PDFX3 2002** (was **PDFX3**)
- **PDFX3 2003**
- **PDFA DRAFT**
- Removed descriptions of legacy Adobe PDF Settings files: **eBook**, **Press**, **Print**, **Screen**, and **CJKScreen**.
- Moved information about using Distiller to combine PostScript files from *Acrobat Guide* online Help to this document.
- Added new parameters information:
 - **CropColorImages** parameter
 - **CropGrayImages** parameter
 - **CropMonolImages** parameter
 - **ColorImageMinResolution** parameter
 - **ColorImageMinResolutionPolicy** parameter
 - **GrayImageMinResolution** parameter
 - **GrayImageMinResolutionPolicy** parameter
 - **MonolImageMinResolution** parameter
 - **MonolImageMinResolutionPolicy** parameter
 - **ColorImageMinDownsampleDepth** parameter
 - **GrayImageMinDownsampleDepth** parameter
 - **EmbedOpenType** parameter
 - **DetectCurves** parameter
 - **PreserveDICMYKValues** parameter
 - **CMYK** value for **ColorConversionStrategy** parameter
 - **1.6** value for **CompatibilityLevel** parameter
 - **PreserveFlatness** parameter
 - **CheckCompliance** parameter
 - **Use Output Condition Identifier** value for **PDFXOutputIntentProfile** parameter
 - **PDFXOutputConditionIdentifier** parameter
 - **ParseICCPProfilesInComments** parameter

This appendix describes how Distiller converts **setpagedevice**-type PostScript key-word pairs and parameters into a JDF file. Distiller creates a JDF file if the **CreateJDFFile** parameter is set to true. The version of JDF created is 1.1 (the specification is available from http://www.cip4.org/documents/jdf_specifications).

This appendix uses XPath expressions to identify specific attributes. *XPath* is a language for addressing parts of an XML document, as defined in XML Path Language (XPath) Version 1.0, which is available from <http://www.w3.org/TR/xpath>. The conventions that appear in the following tables are shown below:

Expression ::= JDFRoot' / Attribute | JDFRoot' / Children' / Attribute

JDFRoot ::= 'JDF'

Children ::= Element | Element' / Children

Element ::= element

Attribute ::= '@'attribute

A.1 Basic JDF File Created

Distiller creates a basic JDF document whose root node is a JDF element with Type="Product". Under that root node, Distiller creates three sub-elements:

- A JDF element with **Type = "Combined"**;
- A ResourcePool element that describes the document produced;
- and an AuditPool element that describes the results of distillation.

The resulting root node is populated with elements that describe the incoming PostScript stream, **Combined** process node, and the following items:

- **setpagedevice**-type operators. Whenever Distiller encounters a supported **setpagedevice**-type operator, it represents the key value as an entry in one of the parameter resources associated with the **ResourceDefinition** process. (Section A.2, "Representing Certain PostScript Keys as JDF Elements and Attributes")
- **DSC comments**. Whenever Distiller encounters certain DSC comments, it represents those comments in the RunList for the PDF file. (Section A.3, "Mapping of Certain DSC Comments into JDF Elements and Attributes")
- **Parameters**. Distiller creates a **PSToPDFConversionParams** resource which it populates with attributes that correspond to the settings of the parameters as of the end of the first page of the job. If the parameter **ColorConversionStrategy** is NOT **LeaveColorUnchanged**, Distiller also creates a **ColorSpaceConversionParams** resource, which it populates as it does for **PSToPDFConversionParams**. (Section A.4, "Mapping of Parameters into JDF Elements and Attributes")

A.2 Representing Certain PostScript Keys as JDF Elements and Attributes

Distiller represents selected **setpagedevice**, **settrapparms**, or **settrapzone** PostScript key-word pairs as JDF entries. It does so by creating and populating corresponding JDF resource elements in a **ResourceDefinition** resource pool, as described in [Table A.1](#).

On occasion, a PostScript key contradicts a Distiller parameter. For information on how this conflict is resolved, see *Using Adobe Normalizer Server, Version 6.0.4*, section 7.2.

TABLE A.1 PostScript keys converted into JDF ResourceDefinition resources

PostScript key	Representation in JDF ResourceDefinition resources
setpagedevice	Distiller converts the setpagedevice key-word pairs into the ResourceDefinition attributes described in Table A.2 . Some keys are omitted from the table because they do not have logical equivalents in the <i>JDF Specification</i> .
settrapparms	Distiller converts the settrapparms key-word pairs into the ResourceDefinition attributes described in Table A.3 .
settrapzone	Distiller converts the settrapzone key-word pairs into the ResourceDefinition attributes described in Table A.4 .

TABLE A.2 Mapping from setpagedevice keys to JDF entries

Key Name	Entry in /JDF / ResourcePool
Collate	<i>DigitalPrintingParams</i> /@ Collate
DeviceRenderingInfo <<ValuesPerColorComponent>>	<i>RenderingParams</i> /@ ColorantDepth
Duplex	<i>LayoutPreparationParams</i> /@ Sides
HWResolution	<i>RenderingParams</i> / ObjectResolution /@ Resolution
Jog	<i>Component</i> / Disjointing /@ OffsetAmount
ManualFeed	<i>DigitalPrintingParams</i> /@ ManualFeed
MediaColor	<i>DigitalPrintingParams</i> / Media /@ MediaColorName
MediaPosition	<i>DigitalPrintingParams</i> / Media / Location /@ LocationName
MediaType	<i>DigitalPrintingParams</i> / Media /@ UserMediaType
MediaWeight	<i>DigitalPrintingParams</i> / Media /@ Weight

TABLE A.2 Mapping from *setpagedevice* keys to JDF entries (Continued)

Key Name	Entry in /JDF / ResourcePool
MirrorPrint	<i>ImageSetterParams</i> /@ <i>MirrorAround</i>
NegativePrint	<i>ImageSetterParams</i> /@ <i>Polarity</i>
NumCopies	If Collate is TRUE, <i>RunList</i> /@ <i>PageCopies</i> . Otherwise, <i>RunList</i> /@ <i>DocCopies</i>
PageSize	<i>DigitalPrintingParams</i> / <i>Media</i> /@ <i>Dimension</i>
ProcessColorModel	<i>ColorantControl</i> /@ <i>ProcessColorModel</i>
SeparationColorNames	<i>ColorantControl</i> /@ <i>ColorantParams</i>
SeparationOrder	<i>ColorantControl</i> /@ <i>ColorantOrder</i>
Separations	<i>ColorantControl</i> /@ <i>ForceSeparations</i>
Trapping	<i>TrappingDetails</i> /@ <i>Trapping</i>
TrappingDetails <<Type>>	<i>TrappingDetails</i> /@ <i>TrappingType</i>
TrappingDetails <<ColorantDetails <<ColorantName>> >>	<i>ColorantControl</i> / <i>ColorPool</i> /@ <i>ColorName</i>
TrappingDetails <<ColorantDetails <<ColorantType>> >>	<i>ColorantControl</i> / <i>ColorPool</i> / <i>Color</i> /@ <i>ColorType</i>
TrappingDetails <<ColorantDetails <<NeutralDensity>> >>	<i>ColorantControl</i> / <i>ColorPool</i> / <i>Color</i> /@ <i>NeutralDensity</i>
TrappingDetails <<TrappingOrder>>	<i>TrappingDetails</i> /@ <i>TrappingOrder</i>
Tumble	<i>LayoutPreparation</i> /@ <i>Sides</i>

TABLE A.3 Mapping from *settrapparms* keys to JDF *TrappingDetails* entries

Key Name	Entry in /JDF / ResourcePool / <i>TrappingDetails</i> / <i>TrapRegion</i> / <i>TrappingParams</i>
BlackColorLimit	@ <i>BlackColorLimit</i>
BlackDensityLimit	@ <i>BlackDensityLimit</i>
BlackWidth	@ <i>BlackWidth</i>

TABLE A.3 Mapping from *settrapparms* keys to JDF *TrappingDetails* entries (Continued)

Key Name	Entry in /JDF / ResourcePool / TrappingDetails / TrapRegion / TrappingParams
ColorantZoneDetails <>StepLimit>>	@ <i>ColorantZoneDetails</i> /@ <i>StepLimit</i>
ColorantZoneDetails <>TrapColorScaling>>	@ <i>ColorantZoneDetails</i> /@ <i>TrapColorScaling</i>
ColorantZoneDetails <>TrapPlacement>>	@ <i>ColorantZoneDetails</i> /@ <i>ADBE</i> ^a : <i>TrapPlacement</i>
Enabled	@ <i>Enabled</i>
ImageInternalTrapping	@ <i>ImageInternalTrapping</i>
ImageMaskTrapping	@ <i>ImageMaskTrapping</i>
ImageResolution	@ <i>ImageResolution</i>
ImageToImageTrapping	@ <i>ImageToImageTrapping</i>
ImageToObjectTrapping	@ <i>ImageToObjectTrapping</i>
ImageTrapPlacement	@ <i>ImageTrapPlacement</i>
ImageTrapWidth	@ <i>ADBE:ImageTrapWidth</i>
MinimumBlackWidth	@ <i>MinimumBlackWidth</i>
SlidingTrapLimit	@ <i>SlidingTrapLimit</i>
StepLimit	@ <i>StepLimit</i>
TrapColorScaling	@ <i>TrapColorScaling</i>
TrapEndStyle	@ <i>TrapEndStyle</i>
TrapJoinStyle	@ <i>TrapJoinStyle</i>
TrapWidth	@ <i>TrapWidth</i>

a. In the JDF document, Distiller defines **ADBE** as the namespace <http://ns.adobe.com/JDF>.

TABLE A.4 Mapping from *settrapzone* keys to JDF *TrappingDetails* entries

Key Name	Entry in /JDF / ResourcePool / TrappingDetails
settrapzone	<i>TrapRegion</i> /@ <i>TrappingZone</i>

A.2.1 Converting Between the Linear Representation of `setpagedevice` Keys and Their Hierarchical JDF Counterparts

The `setpagedevice` keys that appear in PostScript streams/files are presented in a linear fashion. That is, hierarchical relationships are represented by repeating higher level information. In contrast, the JDF format represents a hierarchy.

This section describes how Distiller converts between that linear representation and the JDF hierarchy.

For example, the *JDF Specification* allows the **TrapParams** resource element to appear as a child of the **TrappingDetails** resource element and the **TrapZones** resource element. For example:

FIGURE 1.1 Relationship between JDF TrapParams and other JDF resources

```
<TrappingDetails>
  <TrapRegion>
    <TrapParams .../>
  </TrapRegion.>
</TrappingDetails>
```

Distiller always subordinates **TrapParams** resources to **TrapRegion** resources. That is, Distiller never produces entries, such as the first **TrapParams** resource at left. Instead, if Distiller has set a default trapping zone then it is set for all the pages (using the second **TrapParam**). Subsequently any **settrapzone/settrapparam** settings cause a new **TrapZone** with associated **TrapParams**. There can be many of these per page.

TrapRegions elements (with associated **TrapParams** elements) are created from each **settrapzone** PostScript call using the **trapparams** set at the time (by **settrapparams**) and the **Page** key is set. Default **trapzones** (set as part of an unencapsulated PostScript job as per the *PostScript Language Reference, third edition*) are turned in to a **trapregion** that applies to all pages.

More specifically, the trapping settings may be different for two separate regions of a particular page. For example, the title text and logo of a page might have different settings compared to those used for the body text. A particular image could then also have different settings. As a result, a **TrapZone** is drawn around each object (a normal PostScript path) and different **trapparams** set for each object.

A.3 Mapping of Certain DSC Comments into JDF Elements and Attributes

The presence of the **%%Page** DSC comment in a PostScript stream indicates the beginning of a page in the stream. The presence of the **%%PlateColor** DSC comment in conjunction with **%%Page** indicates the beginning of a pre-separated page for a particular colorant.

Distiller may use the **%%Page** and **%%PlateColor** comments to create a partitioned **RunList** that represents the structure of the full-document PDF file it produces for a PostScript stream, depending on the document structure implied by those comments, as described in the following subsections. The **RunList** is in the **ResourcePool**, which is at the same level as the **Combined** process node.

A.3.1 Composite Jobs

Composite jobs are indicated by the absence of any **%%PlateColor** comments in the PostScript stream.

Normalizer produces un-partitioned **RunLists** for composite jobs. Changes in page device key values are not considered.

A.3.2 Pre-Separated Jobs with Interleaved Separations

Pre-separated jobs with interleaved separations are indicated by the appearance of **%%PlateColor** comments soon after each **%%Page** comment, with each **%%PlateColor** specifying the next colorant in the sequence. That is, the separations that compose a single page appear sequentially, (i.e. cyan separation, magenta separation, yellow separation, and black separation).

When processing pre-separated jobs with interleaved separations, Distiller uses the **%%Page** and **%%PlateColor** DSC comments to create a **RunList** element partitioned on the keys **Run** and **Separation** and to create a **RunIndex** that references the pages in that **RunList**.

Distiller creates an additional **RunIndex** range for the pages that apply to each set of page device key values.

A.3.3 Pre-separated Single-colorant Jobs

Pre-separated single-colorant jobs are the same as [Pre-Separated Jobs with Interleaved Separations](#), except all **%%PlateColor** comments describe a single colorant.

When processing pre-separated single-colorant jobs, Distiller uses the **%%Page** and **%%PlateColor** DSC comments as described for pre-separated jobs with interleaved separations, except the RunList contains a single partition with **Run=1** and **Separation** set to the colorant name provided in **%%PlateColor**.

A.4 Mapping of Parameters into JDF Elements and Attributes

This section describes how Distiller converts parameter settings into JDF element and attribute settings. It presents one section for each category of parameter, as follows:

- [“General” on page 115](#)
- [“Image Compression” on page 116](#)

- “Fonts” on page 119
- “Color Conversion” on page 119
- “Advanced” on page 121

Distiller produces only those JDF attributes described in this section. Some parameters (such as **CreateJobTicket**) do NOT have JDF attribute counterparts. In contrast, some JDF attributes in applicable elements do not correlate with parameters, such as the **RenderingIntent** attribute in the **ColorSpaceConversionParams** element.

Distiller represents parameters as values for the attributes in the following resource elements:

- **PSToPDFConversionParams**
- **FontParams**
- **ImageCompressionParams**
- **ColorSpaceConversionParams**

Distiller does not create the optional¹ **ColorantControl** element.

A.4.1 General

Table A.5 specifies the conversion from Distiller general parameters into JDF elements.

TABLE A.5 Conversion from Distiller general parameters into JDF attributes

Parameter	Attribute name in the PSToPDFConversionParams resource
AutoRotatePages	@ AutoRotatePages
Binding	@ Binding
CompatibilityLevel	@ PDFVersion and ColorSpaceConversionParams /@ Operation Table A.9 describes the role of CompatibilityLevel in deriving the Operation value
CompressObjects	@ADBE:CompressObjects
CoreDistVersion	Not represented in JDF. CoreDistVersion is a read-only parameter that is meaningless in JDF.
DoThumbnails	@ DoThumbnails
EndPage	@ EndPage
ImageMemory	@ ImageMemory

1. Version 1.1a of the JDF Specification changed the **ColorantControl** element in a **PSToPDFConversion** process node from required to optional.

TABLE A.5 Conversion from Distiller general parameters into JDF attributes (Continued)

Parameter	Attribute name in the PSToPDFConversionParams resource
Optimize	<i>@ Optimize</i>
StartPage	<i>@ StartPage</i>

A.4.2 Image Compression

The Distiller image compression parameters map into the JDF **ImageCompressionParams** element, which may have up to three **ImageCompression** subelements, one for each of the following image types:

- Color
- Grayscale
- Monochrome

Each **ImageCompression** subelement contains an **ImageType** attribute that identifies the type of image it represents.

TABLE A.6 Conversion from Distiller Image Compression parameters into JDF ImageCompression subelement

Parameter	Attribute name in the color, grayscale or monochrome ImageCompression subelement
AntiAliasColorImages, AntiAliasGrayImages, or AntiAliasMonImages,	<i>@ AntiAliasImages</i>
AutoFilterColorImages or AutoFilterGrayImages (Not relevant to monochrome images.)	<i>@ AutoFilterImages</i>

TABLE A.6 Conversion from Distiller Image Compression parameters into JDF ImageCompression subelement (Continued)

Parameter	Attribute name in the color, grayscale or monochrome ImageCompression subelement								
AutoFilterColorImages AutoFilterGrayImages /ColorACSIImageDict </QFactor>> /GrayACSIImageDict </QFactor>> /ColorImageDict </QFactor>> /GrayImageDict </QFactor>> /MonolImageDict </QFactor>> Table A.7 describes how the auto filter parameter influences selection of a /QFactor value.	@ DCTQuality Distiller calculates DCTQuality by dividing the selected QFactor by 100. For example:								
	<table border="1"> <thead> <tr> <th>QFactor</th><th>DCTQuality</th></tr> </thead> <tbody> <tr> <td>2.40</td><td>0.0240</td></tr> <tr> <td>1.30</td><td>0.0130</td></tr> <tr> <td>0.15</td><td>0.0015</td></tr> </tbody> </table>	QFactor	DCTQuality	2.40	0.0240	1.30	0.0130	0.15	0.0015
QFactor	DCTQuality								
2.40	0.0240								
1.30	0.0130								
0.15	0.0015								
ColorImageDepth , GrayImageDepth , or /MonolImageDepth	@ ImageDepth								
ColorImageDownsampleThreshold , GrayImageDownsampleThreshold , or MonolImageDownsampleThreshold	@ ImageDownsampleThreshold								
ColorImageDownsampleType , GrayImageDownsampleType , or MonolImageDownsampleType	@ ImageDownsampleType								
ColorImageFilter , GrayImageFilter , or MonolImageFilter	ImageFilter or ADBE:ImageFilter The latter being used to represent the value JPXEncode , LZWEncode , or RunLengthEncode .								
ColorImageResolution , GrayImageResolution , or MonolImageResolution	@ ImageResolution								
JPEG2000ColorACSIImageDict </Quality>> JPEG2000GrayACSIImageDict </Quality>> JPEG2000ColorImageDict </Quality>> JPEG2000GrayImageDict </Quality>>	@ADBE^a:JPXQuality								

TABLE A.6 Conversion from Distiller Image Compression parameters into JDF *ImageCompression* subelement (Continued)

Parameter	Attribute name in the color, grayscale or monochrome <i>ImageCompression</i> subelement
ConvertImagesToIndexed	@ ConvertImagesToIndexed (Represented only in an <i>ImageCompressionParams</i> element with <i>ImageType</i> = “Color”.)
DownsampleColorImages , DownsampleGrayImages , or DownsampleMonolImages	@ DownsampleImages
EncodeColorImages , EncodeGrayImages , or EncodeMonolImages	@ EncodeImages
a. ADBE must be defined as the namespace http://ns.adobe.com/JDF . That is, JDF files that contain elements or attributes that use the ADBE prefix must also contain the definition <i>xmlNs:ADBE="http://ns.adobe.com/JDF"</i> .	

TABLE A.7 Conversion from Parameters into the JDF *DCTQuality* attribute

Condition	Distiller compression dictionary key-word pair used to derive the value of <i>ImageCompression</i> /@ <i>DCTQuality</i>
If AutoFilterColorImages is true	/ColorACSIImageDict </QFactor>
If AutoFilterColorImages is false	/ColorImageDict </QFactor>
If AutoFilterGrayImages is true	/GrayACSIImageDict </QFactor>
If AutoFilterGrayImages is false	/GrayImageDict </QFactor>
If AutoFilterGrayImages value is irrelevant	/MonolImageDict </QFactor>

A.4.3 Page Compression

CompressPages is the sole Distiller page compression parameter. Distiller converts it into the *PSToPDFConversionParams CompressPages* attribute.

A.4.4 Fonts

Distiller converts each Distiller font parameter into the attribute in the JDF **FontParams** resource element with the same name. In other words, for each Distiller font parameter, there is an identically-named attribute in the **FontParams** element.

A.4.5 Color Conversion

If **ColorConversionStrategy** is **LeaveColorUnchanged**, **ColorSpaceConversionParams** element is omitted from the JDF. Otherwise, conversion is as described in [Table A.8](#).

TABLE A.8 Conversion from Distiller color conversion parameters to JDF ColorSpaceConversionParams attributes

Parameter	Attribute name in ColorSpaceConversionParams
CalCMYKProfile Used as the ICC profile FileSpec in the ColorSpaceConversionOp resource that contains Type = “CMYK” .	FileSpec and @ Type
CalGrayProfile Used as the ICC profile FileSpec in the ColorSpaceConversionOp resource that contains Type = “Gray” .	ColorSpaceConversionOp / FileSpec and ColorSpaceConversionOp /@ Type
CalRGBProfile Used as the ICC profile FileSpec in the ColorSpaceConversionOp resource that contains Type = “RGB” .	ColorSpaceConversionOp / FileSpec and ColorSpaceConversionOp /@ Type
ColorConversionStrategy	ColorSpaceConversionOp /@ Operation and ColorSpaceConversionOp /@ SourceObjects , as described in Table A.9 and Table A.10 .
DefaultRenderingIntent	PSToPDFConversionParams /@ DefaultRenderingIntent
PreserveHalftoneInfo	@ PreserveHalftoneInfo
PreserveOverprintSettings	@ PreserveOverprintSetting

TABLE A.8 Conversion from Distiller color conversion parameters to JDF ColorSpaceConversionParams attributes (Continued)

Parameter	Attribute name in ColorSpaceConversionParams
sRGBProfile	FileSpec If ColorConversionStrategy is sRGB , Distiller creates a FileSpec element with Usage="FinalTargetDevice" and a UID value that reflects the ICC profile used for converting color spaces to CalRGB (PDF 1.2) or sRGB (PDF 1.3 and above).
TransferFunctionInfo	@ TransferFunctionInfo
UCRandBGInfo	@ UCRandBG
None; however, Distiller specifies the built-in color management system.	@ ColorManagementSystem

TABLE A.9 Conversion from ColorConversionStrategy into Operation

ColorConversionStrategy value	Operation attribute value
Device independent color ^a and CompatibilityLevel <= 1.2	Convert
Device independent color and CompatibilityLevel > 1.2	Tag
sRGB	Convert
a. ColorConversionStrategy == UseDeviceIndependentColor ColorConversionStrategy == UseDeviceIndependentColorForImages.	

TABLE A.10 Conversion from /ColorConversionStrategy into SourceObjects

ColorConversionStrategy value	SourceObjects attribute value
UseDeviceIndependentColor	All

TABLE A.10 Conversion from /ColorConversionStrategy into SourceObjects

ColorConversionStrategy value	SourceObjects attribute value
sRGB	All and FinalTargetDevice set to sRGB. NOTE: if the conversion is sRGB, then we do NOT create a ColorSpaceConversionOp of SourceCS = Gray because the Gray colors are not changed.
UseDeviceIndependentColorForImages	ImagePhotographic ImageScreenShot

A.4.6 Advanced

Table A.11 specifies the conversion from Distiller advanced parameters into JDF elements.

TABLE A.11 Conversion from Distiller advanced parameters into JDF elements

Parameter	Attribute name in the PToPDFConversion resource
AllowPSXObjects	@ADBE^a:AllowPSXObjects
AllowTransparency	@ADBE:AllowTransparency
ASCII85EncodePages	@ ASCII85EncodePages
AutoPositionEPSFiles	AdvancedParams /@ AutoPositionEPSInfo
CreateJobTicket	Not represented in JDF.
DetectBlends	@ DetectBlend (not a typo)
EmbedJobOptions	@ADBE:EmbedJobOptions
EmitDSCWarnings	AdvancedParams /@ EmitDSCWarnings
LockDistillerParams	AdvancedParams /@ LockDistillerParams
OPM	@ OverPrintMode
ParseDSComments	AdvancedParams /@ ParseDSComments
ParseDSCommentsForDocInfo	AdvancedParams /@ ParseDSCommentsForDocInfo
PassThroughJPEGImages	@ADBE:PassThroughJPEGImages
PreserveCopyPage	AdvancedParams /@ PreserveCopyPage

TABLE A.11 Conversion from Distiller advanced parameters into JDF elements (Continued)

Parameter	Attribute name in the PSToPDFConversion resource
PreserveEPSInfo	AdvancedParams /@ PreserveEPSInfo
PreserveOPIComments	AdvancedParams /@ PreserveOPIComments
UsePrologue	AdvancedParams /@ UsePrologue

a. In the JDF document, Distiller defines **ADBE** as the namespace <http://ns.adobe.com/JDF>.

A.4.7 PDF/X

Table A.12 specifies the conversion from Distiller PDF/X parameters into JDF elements.

TABLE A.12 Conversion from Distiller PDF/X parameters into JDF elements

Parameter	ADBE:PDFXParams attribute name
PDFX1aCheck	@ADBE : PDFX1aCheck
PDFX3Check	@ADBE : PDFX3Check
PDFXCompliantPDFOnly	@ADBE : PDFXCompliantPDFOnly
PDFXNoTrimBoxError	@ADBE : PDFXNoTrimBoxError
PDFXTrimBoxToMediaBoxOffset	PDFXTrimBoxToMediaBoxOffset
PDFXSetBleedBoxToMediaBox	PDFXSetBleedBoxToMediaBox
PDFXBleedBoxToTrimBoxOffset	PDFXBleedBoxToTrimBoxOffset
PDFXOutputIntentProfile	PDFXOutputIntentProfile
PDFXOutputCondition	PDFXOutputCondition
PDFXRegistryName	PDFXRegistryName
PDFXTrapped	PDFXTrapped

A.4.8 Conversion of Parameters Not Available Through UI

All parameters that cannot be set through the UI are converted into attributes in the **ADBE:ThinPDFParams** element, as specified in Table A.13.

TABLE A.13 Conversion from parameters that cannot be set through the Distiller UI

Parameter set using the setdistillerparam key	ADBE:ThinPDFParams attribute name
sidelineEPS	@ ADBE^a:SidelineEPS
filePerPage	@ FilePerPage
sidelineFonts	@ SidelineFonts
sidelineImages	@ SidelineImages

a. In the JDF document, Distiller defines **ADBE** as the namespace <http://ns.adobe.com/JDF>.

Index

A

Adobe PDF settings
 automatic 25
 description of each 55
Adobe PDF settings file
 choosing 11
 advanced parameters 93
 AllowPSXObjects 93
 AllowTransparency 94
 ASCII85EncodePages 94
 AutoPositionEPSFiles 94
 CreateJDFFile 95
 CreateJobTicket 95
 DetectBlends 97
 DetectCurves 96
 DSCReportingLevel 96
 EmbedJobOptions 97
 EmitDSCWarnings 97
 LockDistillerParams 97
 OPM 98
 ParseDSCComments 98
 ParseDSCCommentsForDocInfo 98
 PassThroughJPEGImages 98
 PreserveCopyPage 99
 PreserveEPSInfo 99
 PreserveFlatness 99
 PreserveOPIComments 100
 PreserveOverprintSettings 100
 UsePrologue 100
 AllowPSXObjects 93
 AllowTransparency 94
 AlwaysEmbed 15, 85
 AntiAliasColorImages 61
 AntiAliasGrayImages 70
 anti-aliasing 32
 gray scale images 70
 monochrome images 79
 AntiAliasMonolImages 79
 ASCII85EncodePages 94

AutoFilterColorImages 61
AutoFilterGrayImages 70
automatic Distiller parameters 25
 AutoFilterColorImages 25
 AutoFilterGrayImages 25
 ColorACSIImageDict 25
 ColorImageAutoFilterStrategy 25
 EncodeColorImages 25
 EncodeGrayImages 25
 GrayACSIImageDict 25
 GrayImageAutoFilterStrategy 25
 JPEG2000ColorACSIImageDict 25
 JPEG2000GrayACSIImageDict 25
 AutoPositionEPSFiles 94
 AutoRotatePages 15, 56
 Average 81

B

Bind 48
Binding 15, 56
bit depth, controlling 32
black generation 93

C

CalCMYKProfile 89
CalGrayProfile 55, 89
CalRGBProfile 88
CannotEmbedFontPolicy 85
caveats for setting Distiller parameters 15
CCITTFaxEncode 80
CCITTFaxEncode compression 28, 82
CFF 42
CheckCompliance 15, 101
choosing an Adobe PDF settings file 11
CIDFontType0 fonts 42
CIDFontType1 fonts 42
CIDFontType2 fonts 42
color component 32

- color conversion parameters 88
 - CalCMYKProfile 89
 - CalGrayProfile 89
 - CalRGBProfile 88
 - ColorConversionStrategy 89
 - ColorSettingsFile 89
 - DefaultRenderingIntent 90
 - ParseICCPProfilesInComments 90
 - PreserveDICMYKValues 91
 - PreserveHalftoneInfo 91
 - sRGBProfile 91
 - TransferFunctionInfo 92
- color image compression parameters 61
 - AntiAliasColorImages 61
 - AutoFilterColorImages 61
 - ColorACSIImageDict 62
 - ColorImageAutoFilterStrategy 62
 - ColorImageDepth 63
 - ColorImageDict 63
 - ColorImageDownsampleThreshold 63
 - ColorImageDownsampleType 64
 - ColorImageFilter 65
 - ColorImageMinDownsampleDepth 66
 - ColorImageMinResolution 66
 - ColorImageMinResolutionPolicy 67
 - ColorImageResolution 68
 - ColorSettingsFile 89
 - compact font format 42
 - CompatibilityLevel 15, 16, 56
 - compression
 - CCITTFaxEncode 28, 80, 82
 - customizing for color and grayscale images 26
 - customizing for monochrome images 28
 - Flate 20
 - image quality 26
 - JPEG 20
 - lossless 21
 - lossy 20
 - compression parameters 19
 - automatic Distiller parameters 25
 - color image 61
 - distiller parameters for Flate compression 21
 - grayscale image 69
 - monochrome images 78, 84
 - settings for JPEG compression 20
 - CompressObjects 15, 55, 57
 - CompressPages 84
 - controlling bit depth 32
 - ConvertImagesToIndexed 68
 - copypage 99
 - CoreDistVersion 58
 - CreateJDFFile 95
 - CreateJobTicket 15, 47, 95
 - CropColorImages 68
 - CropGrayImages 71
 - CropMonolImages 79
 - currentdistillerparams 13, 14
 - customizing color and grayscale image compression 26
 - customizing monochrome image compression 28
 - CutMedia 48
- color images
 - setting downsampling threshold for 63
 - setting minimum resolution for downsampling 68
 - setting the downsampling resolution for 29
- color parameters 44
 - UCRandBGInfo 93
- ColorACSIImageDict 62
- ColorConversionStrategy 15, 44, 89
- ColorImageAutoFilterStrategy 62
- ColorImageDepth 63
- ColorImageDict 63
- ColorImageDownsampleThreshold 63
- ColorImageDownsampleType 64
- ColorImageFilter 65
- ColorImageMinDownsampleDepth 66
- ColorImageMinResolution 66
- ColorImageMinResolutionPolicy 67
- ColorImageResolution 68
- ColorSettingsFile 89

D

DCTEncode 25, 63, 65, 72, 75
 DefaultRenderingIntent 90
 definefont 85
 Description 15, 55, 58
 DetectBlends 16, 97
 DetectBlends, use of with idiom recognition 46
 DetectCurves 96
 DeviceRenderingInfo/ ValuesPerColorComponent 48
 Distiller initialization 16
 Distiller parameter, defined 13
 Distiller parameters for Flate compression 21
 AutoFilterColorImages 21
 AutoFilterGrayImages 21
 ColorImageFilters 21
 EncodeColorImages 21
 EncodeGrayImages 21
 EncodeMonolImages 21
 GrayImageFilter 21
 MonolImageFilter 21
 Distiller parameters for JPEG compression
 AutoFilterColorImages 20
 AutoFilterGrayImages 20
 ColorImageDict 20
 ColorImageFilters 20
 Encode ColorImages 20
 EncodeGrayImages 20
 GrayImageDict 20
 GrayImageFilter 20
 Distiller settings for JPEG compression 20
 Distiller-specific operators 13
 distinit.ps 16
 DoThumbnails 15, 58
 DownsampleColorImages 68
 DownsampleGrayImages 71
 DownsampleMonolImages 80
 downsampling and encoding individual images 32
 downsampling and subsampling images
 methods 28
 reasons for 29
 downsampling images
 setting the threshold resolution for 29
 DSCReportingLevel 96

Duplex 48

E

EmbedAllFonts 15, 86
 EmbeddedDocument 99
 EmbedJobOptions 97
 EmbedOpenType 86
 EmitDSCWarnings 97
 EncodeColorImages 69
 EncodeGrayImages 71
 EncodeMonolImages 80
 encoding images 28
 EndPage 59
 EPS files 47
 Example.ps 16

F

filters
 DCTEncode 65, 75
 Flate 65
 FlateEncode 75
 JPXEncode 65, 75
 LZWEncode 65, 75, 82
 RunLengthEncode 82
 findcolorrendering 90
 Flate compression 20
 Flate compression parameters 21
 FlateEncode 65, 75, 82
 Fold 48
 font parameters 41, 84
 AlwaysEmbed 85
 CannotEmbedFontPolicy 85
 EmbedAllFonts 86
 EmbedOpenType 86
 MaxSubsetPct 86
 NeverEmbed 87
 SubsetFonts 88
 fonts
 CIDFontType0 42
 CIDFontType1 42
 CIDFontType2 42
 embedded by Distiller 42

- OpenType 42
- subsetted by Distiller 42
- True Type 42
- Type 1 42
- Type 3 42

- G**
- general parameters 56
 - AutoRotatePages 56
 - Binding 56
 - CompressObjects 57
 - CoreDistVersion 58
 - Description 58
 - DoThumbnails 58
 - EndPage 59
 - HWResolution 59
 - ImageMemory 59
 - Optimize 60
 - PageSize 60
 - StartPage 60
- gray scale images
 - anti-aliasing 70
 - setting minimum resolution for downsampling 78
 - setting the downsampling resolution for 29
- GrayACSDict 72
- GrayImageAutoFilterStrategy 72
- GrayImageDepth 73
- GrayImageDownsampleType 74
- GrayImageFilter 75
- GrayImageMinDownsampleDepth 76
- GrayImageMinResolution 76
- GrayImageMinResolutionPolicy 77
- GrayImageResolution 78
- JPEG2000GrayACSDict 78
- JPEG2000GrayImageDict 78
- grayscale images
 - setting downsampling threshold 73

- H**
- halftone 91
- HWResolution 48, 59

- I**
- IdiomRecognition 46
- image compression parameters 60
- image compression quality 26
- imagemask 27
- ImageMemory 59
- images, encoding 28
- implementing Distiller operators on unsupported PostScript devices 14

- J**
- job ticket keys 47
- Jog 48
- JPEG compression 20
- JPEG compression settings 20
- JPEG2000ColorACSDict 69
- JPEG2000ColorImageDict 69
- JPEG2000GrayACSDict 78
- JPEG2000GrayImageDict 78
- JPXEncode 65, 75

K

keys
 job ticket 47
 key-value pair 13

L

Laminate 48
 LockDistillerParams 15, 16, 97
 lossless compression 21
 lossy compression 20
 LZWEncode 65, 75, 82

M

ManualFeed 48
 MaxSubsetPct 15, 86
 MediaClass 48
 MediaColor 48
 MediaPosition 48
 MediaType 49
 MediaWeight 49
 Metadata 47
 Microsoft Office files 98
 MirrorPrint 49
 mono image compression parameters
 MonolImageDownsampleThreshold 81
 monochrome and grayscale images, caveats for
 processing 33
 monochrome image compression parameters
 MonolImageMinResolution 82
 MonolImageMinResolutionPolicy 83
 monochrome images
 anti-aliasing 79
 customizing compression of 28
 setting downsampling threshold for 81
 setting minimum resolution for downsampling 83
 setting the downsampling resolution for 29
 monochrome images compression parameters 78, 84
 AntiAliasMonolImages 79
 CropMonolImages 79
 DownsampleMonolImages 80
 EncodeMonolImages 80

MonolImageDepth 80
 MonolImageDict 80
 MonolImageDownsampleType 81
 MonolImageFilter 82
 MonolImageResolution 83
 MonolImageDepth 80
 MonolImageDict 80
 MonolImageDownsampleThreshold 81
 MonolImageDownsampleType 81
 MonolImageFilter 82
 MonolImageMinResolution 82
 MonolImageMinResolutionPolicy 83
 MonolImageResolution 83

N

NegativePrint 49
 NeverEmbed 15, 55, 87

O

OpenType fonts 42
 OPM 98
 Optimize 15, 60

P

page compression parameter 84
 CompressPages 84
 PageDirection 56
 PageSize 49, 60
 ParseDSCComments 15, 49, 98
 ParseDSCCommentsForDocInfo 15, 46, 98
 ParseCCProfilesInComments 90
 PassThroughJPEGImages 55, 98
 PDF color space (Out) 44
 PDFX1aCheck 15, 101
 PDFX3Check 15, 102
 PDFXBleedBoxToTrimBoxOffset 15, 103
 PDFXCompliantPDFOnly 15, 102
 PDFXNoTrimBoxError 15, 102
 PDFXOutputCondition 104
 PDFXOutputIntentProfile 15, 103
 PDFXRegistryName 104

PDFXSetBleedBoxToMediaBox 15, 102
PDFXTrapped 15, 104
PDFXTrimBoxtoMediaBoxOffset 15, 102
PostRenderingEnhance 49
PostScript comments and job ticket keys
 relationship 49
PostScript dictionary entry 13
predefined Adobe PDF settings files
 High Quality Printing 11
 PDFA Draft 12
 PDFX1a 2001 12
 PDFX1a 2003 12
 PDFX3 2002 12
 PDFX3 2003 12
 Press Quality 11
 Smallest File Size 12
 Standard 11
PreRenderingEnhance 49
PreserveCopyPage 99
PreserveDICMYKValues 91
PreserveEPSInfo 47, 99
PreserveFlatness 15, 99
PreserveHalftoneInfo 91
PreserveOPIComments 100
PreserveOverprintSettings 100
ProcessColorModel 49
profiles 88
PS color space (In) 44

Q
quality, image compression 26

R
relationship between PostScript comments and job
 ticket keys 49
restore 14
RunLengthEncode 82

S
save 14
SeparationColorNames 49

SeparationOrder 49
Separations 49
setblackgeneration 93
setcolorrendering 90
setdistillerparams 13, 14, 15
setoverprint 100
setpagedevice 13
setpagedevice keys
 Bind 48
 CutMedia 48
 DeviceRenderingInfo/ValuesPerColorComponent
 48
 Duplex 48
 Fold 48
 HWResolution 48, 59
 Jog 48
 Laminate 48
 ManualFeed 48
 MediaClass 48
 MediaColor 48
 MediaPosition 48
 MediaType 49
 MediaWeight 49
 MirrorPrint 49
 NegativePrint 49
 PageSize 49, 60
 PostRenderingEnhance 49
 PreRenderingEnhance 49
 ProcessColorModel 49
 SeparationColorNames 49
 SeparationOrder 49
 Separations 49
 Staple 49
 Trapping 49
 TrappingDetails 49
 Trim 49
 Tumble 49
setting Distiller parameters in PostScript 13
setting Distiller parameters, caveats to observe 15
setting downsampling threshold for color images 63
setting downsampling threshold for gray scale
 images 73
setting downsampling threshold for monochrome
 images 81

setting minimum resolution for downsampling color images 68
 setting minimum resolution for downsampling gray scale images 78
 setting minimum resolution for downsampling monochrome images 83
 setundercolorremoval 93
 sRGBProfile 91
 standards parameters 101
 CheckCompliance 101
 PDFX1aCheck 101
 PDFX3Check 102
 PDFXBleedBoxToTrimBoxOffset 103
 PDFXCompliantPDFOnly 102
 PDFXNoTrimBoxError 102
 PDFXOutputCondition 104
 PDFXOutputIntentProfile 103
 PDFXRegistryName 104
 PDFXSetBleedBoxToMediaBox 102
 PDFXTrapped 104
 PDFXTrimBoxToMediaBoxOffset 102
 Staple 49
 StartPage 60
 stencil masks 27
 SubsetFonts 15, 88

X

XMP Adobe Standard Metadata schema 47

Z

ZIP. See Flate. 20

T
 TransferFunctionInfo 55, 92
 Trapping 49
 TrappingDetails 49
 Trim 49
 True Type fonts 42
 Tumble 49
 Type 1 fonts 42
 Type 3 fonts 42
 typecheck 14

U

UCRandBGInfo 93
 under color removal 93
 UseFlateCompression 105
 UsePrologue 16, 100

