



Acrobat PDF Writer Control Interface Specification

Adobe Developer Support

Technical Note #5159

08 December 1994

Adobe Systems Incorporated

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

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

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

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

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

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

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

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

Adobe, the Adobe logo, Acrobat, and PostScript are trademarks of Adobe Systems Incorporated or its subsidiaries and may be registered in certain jurisdictions. Apple and Macintosh are registered trademarks and QuickDraw is a trademark of Apple Computer, Inc. Microsoft and MS-DOS are registered trademarks and Windows is a trademark of Microsoft Corporation. Other brand or product names are the trademarks or registered trademarks of their respective holders.

This publication and the information herein is furnished AS IS, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies, makes no warranty of any kind (express, implied or statutory) with respect to this publication, and expressly disclaims any and all warranties of merchantability, fitness for particular purposes and noninfringement of third party rights.



Contents

- 1 Introduction 5
- 2 Apple Macintosh 5
- 3 Microsoft Windows 6
 - Windows initialization file (WIN.INI) settings 7
 - Windows messages 9
 - Determining when PDF Writer has begun processing 9
 - Determining when PDF Writer has finished processing 9
 - Windows printer escape function 9

Acrobat PDF Writer Control Interface Specification

1 Introduction

This document describes the Acrobat™ PDF Writer control interface, present in version 2 of the PDF Writer driver for Apple® Macintosh® and Microsoft® Windows™ platforms. The PDF Writer is a system-level QuickDraw™ (Macintosh) or GDI (Windows) printer driver that produces Portable Document Format (PDF) files from almost any application. The control interface allows applications to programatically control the PDF Writer driver.

2 Apple Macintosh

The Macintosh version of PDF Writer supports a C language Application Programming Interface (API). Software that uses this API can:

- Set the output file name of the Portable Document Format (PDF) file that the PDF writer creates.
- Suppress the Page Setup and Print dialog boxes.
- Request that the Acrobat viewer be launched to display the output file immediately after PDF Writer has finished creating that file.

This API is defined in the header file *userinteraction.h* and implemented in *userinteraction.c*. These files are included in the Acrobat Software Development Kits (SDK). Projects that use the PDF Writer API must include *userinteraction.c*. The header file *userinteraction.h* must be included by those source files that use the PDF Writer API.

To control the PDF Writer, members of a structure of type `tUserInter` (defined in *userinteraction.h*) are set to specific values. Table 1 lists the structure members.

Table 1 *Members of the tUserInter structure*

Structure Member	Description
disable	Disable Page Setup and Print dialog boxes (Boolean).
viewPDF	Launch the Acrobat viewer after printing (Boolean).
name	Output (PDF) file name (Str255).

Two functions (prototyped in *userinteraction.h*) are used to get a tUserInter structure and fill its members:

GetUserInteraction

<code>Boolean GetUserInteraction (tUserInterPtr userInter)</code>	
Description	Gets the current PDF Writer user interaction state.
Parameters	<code>userInter</code> The address of a structure of type tUserInter.
Return Value	Returns <code>true</code> if successful, <code>false</code> otherwise.
Example	<pre>Boolean bGotUI; tUserInter userInter; bGotUI = GetUserInteraction(&userInter);</pre>

SetUserInteraction

<code>void SetUserInteraction (tUserInterPtr userInter)</code>	
Description	Sets the PDF Writer user interaction state.
Parameters	<code>userInter</code> The address of a structure of type tUserInter.
Return Value	None
Example	<pre>Boolean bGotUI; tUserInter userInter; bGotUI = GetUserInteraction(&userInter); if (bGotUI) { ui.viewPDF = true; SetUserInteraction(&ui); }</pre>

3 Microsoft Windows

The Windows version of PDF Writer supports Windows messaging, changing settings through the Windows initialization (*WIN.INI*) file, and changing settings using the Windows printer escape function.

3.1 Windows initialization file (WIN.INI) settings

A number of PDF Writer settings can be read from and written to the [Acrobat PDFWriter] section of the Windows *WIN.INI* file. The WriteProfileString(), GetProfileString(), and GetProfileInt() Windows API functions are used for this purpose.

Table 2 *PDF Writer WIN.INI settings*

<i>Key</i>	<i>Type and Values</i>	<i>Default</i>	<i>Description</i>
PDFFileName	path name string	none	Full path name for PDF file.
cpunits	0 = inches, 1 = mm, 2 = points	0	Measurement units used for display of custom page size in print setup dialog box.
cpwidewhole	points integer > 0	612	Document width, whole part.
cpwidepart	1/1000 point integer ≥ 0	0	Document width, fractional part.
cpheightwhole	points integer > 0	792	Document height, whole part.
cpheightpart	1/1000 point integer ≥ 0	0	Document height, fractional part.
cpmarginwhole	points integer ≥ 0	18	Margins, whole part.
cpmarginpart	1/1000 point integer ≥ 0	0	Margins, fractional part.
orient	1 = portrait, 2 = landscape	1	Page orientation.
custom	boolean 0 = no, 1 = yes	0	Custom paper size?
compASCII	boolean 0 = no, 1 = yes	0	Use ASCII-85 filter on entire file?
comptg	boolean 0 = no, 1 = yes	1	Compress text and graphics?
compcolor	boolean 0 = no, 1 = yes	0	Compress color and grayscale images?

<i>Key</i>	<i>Type and Values</i>	<i>Default</i>	<i>Description</i>
compmono	boolean 0 = no, 1 = yes	0	Compress monochrome images?
monotype	0 = CCITT3, 1 = CCITT4, 2 = LZW, 3 = RLE	0	Compression type used for monochrome images.
colorqual	0 = JPEG High, 1 = JPEG Medium-High, 2 = JPEG Medium, 3 = JPEG Medium-Low, 4 = JPEG Low, 5 = LZW	3	Compression type used for color and grayscale images.
res	dots per inch 0 = Screen, 1 = 150, 2 = 300, 3 = 600	2	Resolution used for images.
bDocInfo	boolean 0 = no, 1 = yes	1	Display document info dialog box?
bExecViewer	boolean 0 = no, 1 = yes	0	Launch viewer after creating PDF file?
bAlwaysEmbed	boolean 0 = no, 1 = yes	1	Enable always embed list?
bNeverEmbed	boolean 0 = no, 1 = yes	1	Enable never embed list?
bEmbedAllFonts	boolean 0 = no, 1 = yes	1	Embed all fonts?
EmbedNever_Num	integer > 0	0	Length of never embed list.
EmbedAlways_Num	integer > 0	0	Length of always embed list.

<i>Key</i>	<i>Type and Values</i>	<i>Default</i>	<i>Description</i>
EmbedNever{0...}	font name string	n/a	Name of a font in never embed list.
EmbedAlways{0...}	font name string	n/a	Name of a font in always embed list.

3.2 Windows messages

PDF Writer broadcasts Windows messages to all top-level windows to inform other applications when it has begun and completed processing a PDF file.

Note Some languages do not support receiving Windows messages. Applications written using these languages must instead wait for a few seconds and try to open the PDF file or check to see if PDF Writer is still running to determine when PDF Writer has begun or finished processing.

Determining when PDF Writer has begun processing

A “PDFWriter Start” message is sent when PDF Writer has started processing a file. This message is registered using:

```
UINT msgNum = RegisterWindowMessage("PDFWriter Start");
```

Determining when PDF Writer has finished processing

A “PDFWriter Done” message is sent when PDF Writer has completed processing a file. This message is registered using:

```
UINT msgNum = RegisterWindowMessage("PDFWriter Done");
```

3.3 Windows printer escape function

PDF Writer supports setting of the output file name, author name, and creator name by the Escape() Windows API function.

The definitions necessary to use this mechanism are defined in *pdfwctrl.h*. This file is included in the Acrobat Software Development Kit (SDK). The header file *pdfwctrl.h* should be included by those source files that use the Escape() function to modify PDF Writer behavior.

Members of a structure of type PDFINPUT (defined in *pdfwctrl.h*) are set to change the behavior of PDF Writer. Applications can fill in the structure members listed in Table 3.



Table 3 *Members of the PDFINPUT structure*

Structure Member	Description
outputfilename	Output (PDF) file name (char). If set, this file name will be used and the “Save PDF File As” dialog box will not appear.
username	Author name (char).
creator	Creator name (char).

The printer escape passed to `Escape()` is `PDFFILENAME`. If successful, `Escape()` returns with `PDFINPUT_OK`, otherwise it returns with `PDFINPUT_FAIL`. For example, to set the output file name for PDF Writer:

```
PDFINPUT pdfInput;

/* Fill PDFINPUT structure */
strcpy(pdfInput.outputfilename, "c:\\pdfdocs\\work.pdf");
pdfInput.username = 0; /* Use default */
pdfInput.creator = 0; /* Use default */

/* Send PDFFILENAME printer escape. It is assumed that */
/* hDC represents a device context for PDF Writer. */
if (Escape(hDC, PDFFILENAME, sizeof(pdfInput), (LPSTR) &pdfInput,
  NULL) == PDFINPUT_OK) {
  /* Escape successful. Now start print job. */
} else {
  /* Escape failed. */
}
```

The `PDFFILENAME` printer escape should be sent prior to calling the `StartDoc()` Windows API function or using the `STARTDOC` printer escape to start a PDF Writer print job.