

Adobe Acrobat 7.0.5




Acrobat Distiller API Reference

October 3, 2005



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



Copyright 2005 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. Verity is a registered trademark of Verity, Incorporated. UNIX is a registered trademark of The Open Group. Verity is a trademark of Verity, Inc. Lextek is a trademark of Lextek International. 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	5
Contents	5
Other Useful Documentation	5
Conventions Used In This Document	6
Chapter 1 Acrobat Distiller API for Macintosh	7
Apple Event Support.	7
Objects	8
Events	9
Chapter 2 Acrobat Distiller API for Windows	13
Windows Messaging.	13
API to Tell Distiller to Begin Processing and To Send Confirmations	14
API to Determine Distiller Version Number.	15
Distiller Command Line Options	16
How to Turn Off Prompting for Filename	19
UI to Turn Off Prompting for Current User Only	19
UI to Turn Off Prompting for All Users	20
InstallShield Tuner for Adobe Acrobat.	20
Programmatic Control	20
Chapter 3 Acrobat Distiller Automation	21
Introduction	21
Distiller Automation	21
Example.	21
Contents	22
Automation Methods	22
Automation Properties	24
Automation Events.	24



Preface

The Adobe® Acrobat® Distiller® application converts PostScript language page descriptions into Portable Document Format (PDF) files. This technical note describes the Acrobat Distiller control interface, present in version 7 of Acrobat Distiller for the Macintosh and Microsoft Windows platforms. This interface allows other applications to programmatically control the Distiller program. It consists of Apple events on the Macintosh, and Windows messages and command line options on Windows.

Contents

This technical note covers two main topics:

- Distiller API methods for [Apple Event Support](#) and [Windows Messaging](#). The Macintosh subsection describes the Distiller API methods available through AppleScript. This section also discusses the Distiller API methods that are used with Windows messaging.
- [Acrobat Distiller Automation](#). (Acrobat 4.0 and higher versions.) This section describes the Automation methods available for Distiller.

Other Useful Documentation

You should be familiar with the Portable Document Format (PDF).

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 wish to produce PDF files directly.

PostScript Language Reference, third edition. Your reference to the syntax and semantics of the PostScript language and the Adobe imaging model.

Distiller Parameters. Your reference to the Distiller-specific operators that can be placed in the PostScript language input file.

pdfmark Reference. Your reference to 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.

Conventions Used In This Document

Font	Used for	Examples
monospaced	Paths and filenames	<code>C:\templates\mytmpl.fm</code>
	Code examples set off from plain text	These are variable declarations: <code>AVMenu commandMenu,helpMenu;</code>
monospaced bold	Code items within plain text	The <code>GetExtensionID</code> method ...
	Parameter names and literal values in reference documents	The enumeration terminates if <code>proc</code> returns <code>false</code> .
monospaced italic	Pseudocode	<code>ACCB1 void ACCB2 ExeProc(void)</code> <code>{ do something }</code>
	Placeholders in code examples	<code>AFSimple_Calculate(cFunction, cFields)</code>
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 code items within this document	Test whether an ASAtom exists.
bold	PostScript language and PDF operators, keywords, dictionary key names	The <code>setpagedevice</code> operator
	User interface names	The File menu
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

1

Acrobat Distiller API for Macintosh

Apple Event Support

The Macintosh version of the Distiller program supports Apple events. Apple events can be used from programming languages such as C or from AppleScript. Because AppleScript is more straightforward, using it is the recommended way to use Apple events with the Distiller program.

Distiller supports the following Apple events:

- The **Distill** command distills one or more PostScript® language files, creating a Portable Document Format (PDF) file for each.
- The **run** command launches Distiller.
- The **quit** command quits Distiller.

Distiller supports the **application** object.

Each piece of AppleScript example code in this document assumes that it is being executed within an appropriate *tell – end* construct, for example:

```
tell application "Acrobat Distiller 6.0"  
    [Your code]  
end tell
```

Optional items are enclosed in [square brackets].

Objects

application

Description

Distiller's top-level scripting object

Elements

`Document`, `Window`

Properties

Property	Class	Description
<code>postScriptVersion</code>	Unicode text [r/o]	PostScript interpreter version; for example, "3015.102"
<code>locale</code>	Unicode text [r/o]	3-character language code for Distiller's user interface (for example, "ENU" is English)
<code>frontmost</code>	boolean [r/o]	Whether Distiller is the active application.
<code>name</code>	Unicode text [r/o]	The name of the application
<code>version</code>	Unicode text [r/o]	The version of the applicaiton.

Events

Distill

Description

Distills a file.

AppleScript Syntax

```
Distill sourcePath Unicode text [destinationPath Unicode text]
[adobePDFSettingsPath Unicode text]
```

AppleScript Parameters

sourcePath	POSIX path to the input file (the file to be distilled).
destinationPath	POSIX path to the output file. If not specified, the PDF file will be generated in the same folder as the input file.
adobePDFSettingsPath	Either POSIX path to the Adobe PDF Settings file, or the name of one of the settings files in Distiller. If not specified, the settings file selected in the application is used.

Return Value

A boolean indicating status. If **true**, the command succeeded.

Examples

```
Distill sourcePath "hello.ps"
```

```
Distill sourcePath "hello.ps" destinationPath "/Users/Me/Desktop"
      adobePDFSettingsPath "PDFX1a"
```

```
Distill sourcePath "/hello.ps" adobePDFSettingsPath "/Users/Shared/Adobe
      PDF 6.0/Settings/PDFX1a.joboptions"
```

The following example uses application properties to determine which settings file is used:

```
set distSetting to "Standard"
if (postScriptVersion as string) is "3015.102" then
    set distSetting to "High Quality"
end if
if locale is "CHT" then
    set distSetting to "Smallest File Size"
end if
Distill sourcePath "/hello.ps" adobePDFSettingsPath distSetting
```

quit

Description

Terminates the Distiller program.

AppleScript Syntax

```
quit
```

Parameters

None

Return Value

None

Example

```
quit
```

run**Description**

Launches the Distiller program and invokes its standard startup procedures.

AppleScript Syntax

```
run
```

Parameters

None

Return Value

None

Example

```
run
```


2

Acrobat Distiller API for Windows

Windows Messaging

The Microsoft Windows version of the Distiller program supports Windows messages that:

- Specify the file or files to process and the output destination.
- Confirm when each file specified in the first Windows message has completed processing.
- Determine the Distiller version number.

The enumerated data types and constants necessary to use the Windows messaging portion of the Distiller control interface are defined in the file `distctrl.h`. This file is included in the Acrobat Software Development Kit (SDK). Source files that use the Distiller control interface must include `distctrl.h`.

Members of a structure of type `DISTILLRECORD` (defined in `distctrl.h`) are set to specify the list of files to distill and the destination for the output files. Applications should fill in the structure members listed in [Table 2.1](#).

TABLE 2.1 *Members of the DISTILLRECORD structure*

Structure Member	Description
<code>param</code>	One of the values listed in Table 2.2 .
<code>fileList</code>	Comma-delimited list of files to distill (char).
<code>outputFile</code>	Destination path for output file or files (char). If multiple files are specified for <code>fileList</code> , specify a directory rather than a specific file, or each destination file will be overwritten by the next.

One of the options listed in [Table 2.2](#) must be specified in the `SendMessage` call to control the appearance of the **Save** dialog box when each source file is processed.

TABLE 2.2 *EnqueueOption constants*

Constant	Description
<code>EQ_NO_SAVE_DIALOG</code>	Do not display the Save dialog box.
<code>EQ_DEFAULT_OLD_DEST</code>	Display the Save dialog box. Use the most recent destination directory as the default destination directory.

TABLE 2.2 *EnqueueOption constants*

Constant	Description
EQ_DEFAULT_SOURCE	Display the Save dialog box. Use the source file's directory as the default destination directory.

API to Tell Distiller to Begin Processing and To Send Confirmations

To tell the Distiller program to begin processing the file list, a **COPYDATASTRUCT** containing the **DM_DISTILL** message and a pointer to a filled out **DISTILLRECORD** needs to be sent to the Distiller program. The following example tells the Distiller program to process a PostScript language file and store the resulting PDF file in a specified directory, omitting the unnecessary Save dialog box.

```

DISTILLRECORD dr;          /* from distctrl.h */
COPYDATASTRUCT cds;
BOOL ok;
LRESULT                rtn;
WORD                   res=0;
char                   msg[ 80];
hinst = ShellExec( NULL, "acrodist.exe", strCmdArgs, NULL, SW_SHOW);
if (res<32){
    sprintf(msg, "WinExec failed: error code = %d", res);
    return;
}
CWnd *hDistillerCWnd = FindWindow("Distiller", NULL);
if (hDistillerCWnd != NULL)
{
    strcpy(dr.outputFile, "c:\\ OUT.PDF");
    strcpy(dr.fileList, "C:\\ TEST.PS");
    dr.param = EQ_NO_SAVE_DIALOG; /* from distctrl.h */
    cds.dwData = DM_DISTILL;
    cds.cbData = sizeof(DISTILLRECORD);
    cds.lpData = (PVOID)&dr;
    ok = (BOOL)hDistillerCWnd->SendMessage(WM_COPYDATA,
        (WPARAM)m_hWnd, (LPARAM)&cds);
    if (ok)
        /* wake up Distiller */
        hDistillerCWnd->PostMessage(WM_TIMER, ID_TIMER, 0L);
}

```

If the sending application is specified in the **WPARAM** parameter of the **SendMessage** call, a **WM_COPYDATA** message is returned to the application after each file specified has been distilled. The **LPARAM** parameter of this **WM_COPYDATA** message will contain a **COPYDATASTRUCT** that will include a Distiller **DM_DONE** message and a pointer to a structure of type **DISTILLRECORD**. The **fileList** member of the structure contains the name of the PostScript language file that was processed, and the **outputFile** member contains the name of the resulting PDF file.

API to Determine Distiller Version Number

An application can get the Distiller program version number by using the Windows Messaging portion of the Distiller control interface, which is defined in `distctrl.h`. Source files that use the Distiller control interface must include `distctrl.h`. To get the Distiller program version number, send a `COPYDATASTRUCT` containing the `DM_VERSION` message to the Distiller program. The Distiller version number will be returned in the return value. The high word has the major version information and the low word has the minor version information.

The following example shows how to obtain the Distiller program version number:

```
COPYDATASTRUCT cds={0};
DWORD dwVersion;
CWnd *hDistillerCWnd = FindWindow("Distiller", NULL);

if (hDistillerCWnd != NULL)
{
    cds.dwData = DM_VERSION;
    cds.cbData = 0;
    cds.lpData = NULL;
    dwVersion = (DWORD)hDistillerCWnd->
SendMessage(WM_COPYDATA, NULL, (LPARAM) &cds);
    if (HIWORD(dwVersion) >= 6)
        ; //6.x and above; LOWORD(dwVersion) has the minor version number.
    else
        ; // Distiller 5.x and below
}
```

The `distctrl.h` file for Windows 32-bit is different from the one for Windows 16-bit since the `WM_COPYDATA` message and `COPYDATASTRUCT` structure are not defined for Windows 16-bit. Make sure you use the correct `distctrl.h` for your application.

In addition to the old `WM_COPYDATA` interface that uses the awkward and limited `DISTILLRECORD` structure, there is the `WM_COPYDATA` interface that simply lets you pass a command line. This way you can use any of the command line options, such as `/J` to specify the Adobe PDF settings file (which the old `WM_COPYDATA` interface does not allow).

Using the new `WM_COPYDATA` interface is similar to the old, except you create a command line string instead of a data structure. An example:

```
char szCmdLine[] = "/O outfile.pdf /J myoptions.joboptions infile.ps"
COPYDATASTRUCT cds;
cds.dwData = DM_CMDLINE;
cds.cbData = strlen(szCmdLine) + sizeof(char);
cds.lpData = szCmdLine;
SendMessage( hwndDistiller, WM_COPYDATA,
             (WPARAM) hwndMyWindow, (LPARAM) &cds );
```

(New message for Distiller 6.0) To get the Distiller program version number, a **COPYDATASTRUCT** containing the **DM_VERSION** message is sent to the Distiller program. The following example tells the Distiller program to give the version information.

```
COPYDATASTRUCT cds = {0};
cds.dwData = DM_VERSION;
DWORD dwVersion = SendMessage( hwndDistiller, WM_COPYDATA,
(WPARAM) hwndMyWindow, (LPARAM) &cds );
```

Distiller Command Line Options

Distiller 6.0 and above support the above command line and **WM_COPYDATA** interfaces and add the **--deletelog** and **--nosecurity** switches.

The Distiller program supports options passed as arguments on the command line. The command line is accessed from the Windows 98, Windows NT, Windows 2000, and WinXP **Start->Run** menu option or from another Windows program using the **WinExec** function. Distiller also supports the use of a Windows command prompt application to access and use the Distiller command line options.

The syntax of the Distiller 7.0 command line is:

```
acrodist [switches] [inputFiles]
```

If any **switches** are present, they must come before any input files. Switches and input filenames are both optional. The command **acrodist** by itself runs Distiller, or if a normal instance of Distiller is already running, it brings Distiller to the foreground.

The **switches** parameter is a list of any of the optional commands. Either the - or / character can begin a switch, which is identified by a single case-independent letter. (Note, however, that the **deletelog** and **nosecurity** switches can only be preceded by "--") There should be a space after the switch letter, and if the switch takes a parameter, another space after the parameter. (The parameter should be in quotes if it contains any spaces.) Do not try to combine switches—give each one its own - or / prefix.

The parameter **inputFiles** is a list of filenames, separated by spaces or commas. Spaces and commas are both legal filename characters; if a filename contains spaces or commas, it should be enclosed in double quotes.

To process a list of PostScript language files:

```
acrodist [switches] inputFiles[, inputFiles...]
```

In [Table 2.3, "Command Line Switches"](#), the optional command line switches are listed.

TABLE 2.3 *Command Line Switches*

Switch	Description
--deletelog:on	Forces Distiller to create or delete the log file after the PDF file is created. --deletelog:on turns on logging for the generated PDF. --deletelog:off turns off logging for the generated PDF.
--deletelog:off	

TABLE 2.3 Command Line Switches

Switch	Description
<code>/E [pdfSettingsFilePath]</code>	<p>Opens Distiller's Adobe PDF Settings dialog to edit the specified Adobe PDF settings file. If you give the filename with no path, Distiller looks in its Settings folder for the file. If you omit the filename, Distiller uses the current default settings file.</p> <p>This switch may not be combined with any other command line options. Distiller will not process any PostScript files and does not display its main window. It just opens the Adobe PDF Settings dialog and exits when you close that window.</p> <p>Put quotes around Adobe PDF settings filenames that contain spaces. For example, the following work:</p> <p>"High Quality" "High Quality.joboptions"</p>
<code>/F</code>	<p>(Distiller 5.0 and higher) Restricts the PostScript file operators (see Section 8.1, "Operator Summary," page 518, in the <i>PostScript Language Reference, third edition</i>) to read-only access to all directories and files in the Distiller installation directory with the exception that write access is allowed in the temporary and hostfont cache directories. For example, if Distiller's installation directory is:</p> <p>C:\Programs\Adobe\Acrobat5.0\Distiller\</p> <p>then the temporary directory is:</p> <p>C:\Programs\Adobe\Acrobat5.0\Distiller\Tmp\ #{PID}</p> <p>where {PID} is the process ID of Distiller, and the hostfont cache directory is:</p> <p>C:\Programs\Adobe\Acrobat5.0\Distiller\Cache\</p> <p>Without using this <code>/F</code> command line switch, there is no change in functionality: a PostScript program will be able to freely use all the file operators.</p>
<code>/J [pdfSettingsFilePath]</code>	<p>Uses <code>pdfSettingsFilePath</code> as the Adobe PDF settings file for any input files specified on the command line. If you give the filename with no path, Distiller looks in its Settings folder for the file. If you omit the filename, Distiller uses the current default settings file. Does not affect any of the Adobe PDF settings you can set with the UI.</p> <p>Put quotes around job option filenames that contain spaces. For example, the following work:</p> <p>"High Quality" "High Quality.joboptions"</p>

TABLE 2.3 Command Line Switches

Switch	Description
<code>/N</code>	<p>Runs a new instance of Distiller, even if another Distiller is already running. (Without this switch, Distiller switches to any “normal” previous instance.) The Distiller instance created with this switch does not process watched folders and is marked so that it will not get activated by other “normal” Distillers that get launched later. The number of new instances of Distiller that can be created with this switch is limited by system resources.</p> <p>NOTE: When using this switch, do not use the DOS start command with the /wait option. The DOS command “start /wait” negates the effect of using /N with Distiller.</p>
<code>--nosecurity</code>	<p>Suppresses the Confirm Security dialog box. No security will be applied to the file.</p> <p>NOTE: This option was implemented only for the Adobe PDF printer and should be passed as a WM_COPYDATA message or via the COM interface. If it is used as a command line option, then the PDF file (from the command line PostScript stream) will not have any security. But if the user drags and drops another PostScript file from the UI, then the security settings in the registry will be applied; this is the reason to show the Confirm Security dialog.</p>
<code>/O outputFileOrFolderPath</code>	<p>Specifies the output PDF file name or a directory name to put PDF files in. If you give a directory name, Distiller uses the input filename for each input file. To process a PostScript language file and name the output PDF file:</p> <pre>acrodist /o destFile sourceFile</pre> <p>To process a list of PostScript language files and place the output PDF files in a specified directory:</p> <pre>acrodist /o directory sourceFile1[, sourceFile2...]</pre>

TABLE 2.3 Command Line Switches

Switch	Description
<code>/Q (:seconds)</code>	<p>Tells Distiller to exit immediately when it becomes idle. Distiller checks this switch only if Distiller is launched as new instance (<code>/N</code>); otherwise idle time is limited by the watched folder timer.</p> <p>If <code>/N</code> is also used, quits Distiller after processing all files on the command line. If <code>/N</code> is not used, quit Distiller after processing all PostScript language files in watched directories and any that were specified on the command line.</p> <p>To force the Distiller program to terminate when it has finished distilling all PostScript language files in watched directories and any that were specified on the command line:</p> <pre>acrodist /q [sourceFile1[, sourceFile2...]]</pre> <p><code>/Q</code> also accepts an optional timeout value in seconds, "<code>/Q:n</code>". Use the timeout to tell Distiller to wait until <code>n</code> seconds of idle time before exiting. <code>n</code> must be a positive integer and cannot be greater than 2147483 seconds (<code>0x7fffffff/1000</code>), or about 24.8 days. The default watched folder timer is set to 10 seconds, so every 10 seconds Distiller wakes up to check the watched folder. Since it's checking every 10 seconds, it's idle for no more than 10 seconds. Also, if the <code>SetTimer</code> value is less than 10 seconds, then the system rounds this value to 10 seconds. Refer to Microsoft API reference for more information.</p>

How to Turn Off Prompting for Filename

Acrobat Distiller prompts for output filenames based on settings that can be managed via a graphical user interface or by programmatic control. The methods include:

UI to Turn Off Prompting for Current User Only

Prompting can be turned off for the current user for all Adobe PDF document creation regardless of the application by following these procedures. First, open the Adobe PDF printer (**Start > Settings > Printers and Faxes > Adobe PDF**), then open the **Adobe PDF Properties** dialog (**Printer > Properties**), click on the **General** tab and then the **Printing Preferences...** button. Click on the **Adobe PDF Settings** tab and change the **Adobe PDF Output Folder** drop-down menu selection from **Prompt for Adobe PDF filename** to **My Documents*.pdf** or click on the **Browse...** button to select a different folder.

UI to Turn Off Prompting for All Users

Prompting can be turned off for all users for all Adobe PDF document creation regardless of the application by following these procedures. First, open the Adobe PDF printer (**Start > Settings > Printers and Faxes > Adobe PDF**), then open the **Adobe PDF Properties** dialog (**Printer > Properties**), click on the **Advanced** tab and then the **Printing Defaults...** button. Click on the **Adobe PDF Settings** tab and change the **Adobe PDF Output Folder** drop-down menu selection from **Prompt for Adobe PDF filename** to **My Documents*.pdf** or click on the **Browse...** button to select a different folder.

NOTE: Administrator privileges are required to take this action.

InstallShield Tuner for Adobe Acrobat

Prompting can be turned off with customized deployment options made available through InstallShield Tuner for Acrobat. Visit the web page <http://www.adobe.com/products/acrobat/deployment.html> for more details.

Programmatic Control

Prompting can be turned off programmatically by modifying the Windows registry. This method applies to the creation of only one PDF document by a specific application for the current user. To do this, entries are added under the Windows registry key:

HKEY_CURRENT_USER\Software\Adobe\Acrobat Distiller\PrinterJobControl

This key takes as subkeys:

(Default)
application

The **(Default)** entry is reserved for possible future use and should not be used. The *application* subkey is the full path of the application that is to have prompting turned off. The value of the *application* subkey is a **REG_SZ** value that is the full path of the output file. For example, the following registry script would turn off prompting for the next printing performed by **wordpad.exe**, printing to the file **c:\MyPDFoutputFileName.pdf**:

```
Windows Registry Editor Version 5.00
[HKEY_CURRENT_USER\Software\Adobe\Acrobat Distiller\PrinterJobControl]
"C:\Program Files\Windows NT\Accessories\wordpad.exe" =
"c:\MyPDFoutputFileName.pdf"
```

NOTE: Though the programming language may require that your backslashes are escaped (for example, "c:\\MyPDFoutputFileName.pdf"), the value of the registry entry must end up with just a single slash as shown above. This key, once established, remains until used and is removed once the Windows API function **StartDoc (HDC hdc, CONST DOCINFO* lpdi)** has successfully completed. Also note that the output folder path must already exist with read and write access for the current user and the destination file must not exist.

3

Acrobat Distiller Automation

Introduction

The previous chapters in this document described two ways to automate Distiller:

- **WM_COPYDATA** interface
- Command line interface

The **WM_COPYDATA** interface provides compatibility for older applications only. It works exactly as before, and you cannot use any of Distiller's new features such as Adobe PDF settings files and input piping through it.

For new applications, the best way to automate Distiller under Microsoft Windows is through the new Automation interface also known as OLE Automation (commonly referred to as ActiveX or COM). The Automation interface makes it easy to create and control a Distiller from any programming language that supports Automation. Distiller supports programming environments written in:

- Visual Basic
- Visual C++ with and without MFC

Distiller Automation

Distiller exposes one interface: **PdfDistiller**. This interface provides methods, properties and events. Unless noted, all examples use Visual Basic notation.

Example

In Visual Basic, if you want to just create and use a Distiller without spooling or events, the code can be as simple as:

```
Dim pdf As PdfDistiller
pdf.FileToPdf "My Test File.ps", "", ""
```

Contents

The remainder of this chapter contains the following reference sections:

- [Automation Methods](#). This section describes each Automation method, including its parameters, return value, and related methods.
 - [Automation Properties](#). This section details the properties that can be set in the various objects. Each property describes the key, the property type (for example, read-only), and the semantic.
 - [Automation Events](#). As it processes each job, Distiller fires several events to notify you of its progress. This section describes these events.
-

Automation Methods

Create

Description

Creates a Distiller instance. You do not need to call this method; a Distiller instance is always created if one of the other methods needs it. You may want to call this if you are handling events and want to display Distiller's startup messages before you submit any jobs.

Each user of the Automation interface gets its own Distiller instance. There is no sharing of a common Distiller as was done with Distiller 3.0 and the **WM_COPYDATA** interface.

Parameters

None

Return Value

None

Exceptions

None

FileToPDF

Description

Submits a PostScript file job to the Distiller.

Parameters

<code>strInputPostScript</code>	The PostScript file to process.
<code>strOutputPDF</code>	The name of the PDF filename.
<code>strPDFOptions</code>	The name and path of the Adobe PDF settings file to use.

Return Value

`short int` (`true` on success, `false` otherwise — if `0`, the parameters are invalid; if `-1`, the PDF creation itself failed). If the user set the `bSpoolJobs` flag before calling this method, then it returns an error only for invalid parameters.

Exceptions

None

FileToPDF2

Description

Submits a PostScript file job to the Distiller. Same as [FileToPDF](#) except for the addition of an option to apply security.

Parameters

<code>strInputPostScript</code>	The PostScript file to process.
<code>strOutputPDF</code>	The name of the PDF filename.
<code>strPDFOptions</code>	The name and path of the Adobe PDF settings file to use.
<code>long bApplySecurity</code>	Boolean that, if greater than <code>0</code> , causes security to be applied to the PDF as currently defined in the Distiller application security dialog.

Return Value

`short int` (`true` on success, `false` otherwise — if `0`, the parameters are invalid; if `-1`, the PDF creation itself failed). If the user set the `bSpoolJobs` flag before calling this method, then it returns an error only for invalid parameters.

Exceptions

None

Automation Properties

bShowWindow

[get/set] As Long

Description

Specifies whether Distiller opens with the status windows. This property takes effect only if you set it before calling the [Create](#) method or any other method. If you have already started Distiller, **bShowWindow** has no effect.

bSpoolJobs

[get/set] As Long

Description

Specifies whether Distiller queues PDF jobs through its internal spooler or processes each job immediately.

By default, **bSpoolJobs** is **false**, and **FileToPDF** processes the PDF job immediately and does not return until the PDF file is created.

If **bSpoolJobs** is **true**, **FileToPDF** submits the PDF job to Distiller's internal job queue and returns immediately. The job will be processed at some later time. To find out when the job is done, you can watch for the events Distiller runs during job processing.

Automation Events

OnJobStart

Run once when a job begins processing.

```
OnJobStart( ByVal strInputPostScript As String,  
            ByVal_ strOutputPDF As String )
```

OnJobDone

Run once when a job completes successfully.

```
OnJobDone( ByVal strInputPostScript As String,  
            _ByVal strOutputPDF As String )
```

OnJobFail

Run once when a job ends unsuccessfully.

```
OnJobFail( ByVal strInputPostScript As String,  
_ByVal strOutputPDF As String )
```

OnLogMessage

Run at various times with the text messages that normally appear in Distiller's message log window. The text contains the usual carriage return-line feed pair at the end of each line. A single call to **OnLogMessage** may contain multiple lines or partial lines of text.

```
OnLogMessage( ByVal strMessage As String )
```

In the current version of Distiller, the text that is passed in this message may contain line feed characters without carriage return characters. The application should not make any assumptions about how this text is formatted and should be prepared to receive either line feed characters (LF) alone or carriage return - line feed (CR-LF) pairs.

OnPercentDone

Run periodically during a job to indicate overall progress.

```
OnPercentDone( ByVal nPercentDone As Long )
```

OnPageNumber

Run periodically during a job to indicate the current page number.

```
OnPageNumber( ByVal nPageNumber As Long )
```

