Contents

1 Acrobat Distiller API Reference

Restriction on Directory Access

Beginning with Acrobat 8.1, Distiller® restricts the directories that PostScript® file operators can access. The new default behavior limits directory access to the temp and font cache directories. Earlier versions of Acrobat DC Distiller allowed PostScript file operators to have unlimited directory access.

The following Acrobat DC Distiller command line options (Windows) and user preference (Mac) enable unlimited directory access. Such unlimited access can pose security problems.

- Windows: -F command line option
- Mac OS: AllowPSFileOps user preference

For information on using these Acrobat DC Distiller command line options and preferences with the EMBED pdfMark command, refer to the *pdfmark Reference*.

Windows Automation

For most applications, the best way to automate Distiller under Windows is through the automation interface. The automation interface makes it easy to create and control Distiller from any programming language that supports automation. Distiller supports Visual Basic and Visual C++ with or without MFC.

Distiller exposes one interface: `PdfDistiller`. This interface provides methods, properties, and events.

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

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

Unless otherwise noted, all examples in this chapter use Visual Basic notation.

Methods

Create

FileToPDF

FileToPDF2

Create

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

Each user of the automation interface has its own Distiller instance. There is no sharing of a common Distiller as is done with the `WM_COPYDATA` interface.

FileToPDF

Submits a PostScript file job to Distiller.

This method always creates a log file, regardless of the setting of the Delete Log Files for Successful Jobs preference in Distiller.

Parameters

`strInputPostScript`	The PostScript file to process.
`strOutputPDF`	The name of the PDF file.
`strPDFOptions`	The name and path of the Adobe PDF settings file to use.

Returns

`short int` (`true` on success, `false` otherwise)

If `0` is returned, the parameters are invalid; if `-1` is returned, 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.

FileToPDF2

Description

Submits a PostScript file job to Distiller. This method is the same as `FileToPDF` except for the addition of an option to apply security.

Parameters

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

Returns

`short int` (`true` on success, `false` otherwise)

If `0` is returned, the parameters are invalid; if `-1` is returned, 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.

Properties

bShowWindow

bSpoolJobs

bShowWindow

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.

Syntax

```
[get/set] As Long
```

bSpoolJobs

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.

Syntax

```
[get/set] As Long
```

Events

OnJobStart

OnJobDone

OnJobFail

OnLogMessage

OnPercentDone

OnPageNumber

OnJobStart

Run once when a job begins processing.

Syntax

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

OnJobDone

Run once when a job completes successfully.

Syntax

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

OnJobFail

Run once when a job ends unsuccessfully.

Syntax

```
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.

A single call to `OnLogMessage` may contain multiple lines or partial lines of text. In the current version of Distiller, the text may contain line feed characters without carriage return characters. Your 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.

Syntax

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

OnPercentDone

Run periodically during a job to indicate overall progress.

Syntax

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

OnPageNumber

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

Syntax

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

2 Windows Messaging

Distiller supports Windows messages that can perform the following tasks:

- 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 Windows messaging are defined in the file `distctrl.h`. This file is included in the Adobe Acrobat DC SDK. Source files that use the Distiller control interface must include `distctrl.h`.

Note: The `WM_COPYDATA` interface provides compatibility for older applications only. You cannot use any of Distiller's newer features with it, such as Adobe PDF settings files and input piping.

Specifying input and output files

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 the following table.

Members of the DISTILLRECORD structure

Structure member	Description
`param`	One of the values listed in the table "EnqueueOption constants" on page 6.
`fileList`	Comma-delimited list of files to distill (char).
`outputFile`	Destination path for output file or files (char). If multiple files are specified for `fileList`, specify a directory rather than a specific file, or each destination file will be overwritten by the next.

Determining use of the Save dialog box

One of the options listed in the following table must be specified in the `SendMessage` call to control the appearance of the Save dialog box when each source file is processed.

EnqueueOption constants

Constant	Description
`EQ_NO_SAVE_DIALOG`	Do not display the Save dialog box.
`EQ_DEFAULT_OLD_DEST`	Display the Save dialog box. Use the most recent destination directory as the default destination directory.
`EQ_DEFAULT_SOURCE`	Display the Save dialog box. Use the source file's directory as the default destination directory.

Processing the file list

To instruct Distiller to begin processing the file list, a `COPYDATASTRUCT` containing the `DM_DISTILL` message and a pointer to a filled out `DISTILLRECORD` is sent to Distiller. The following example instructs Distiller to process a PostScript language file and store the resulting PDF file in a specified directory, omitting the unnecessary Save dialog box.

Example: *Processing a file list and omitting the 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.

Determining the version number

To determine the Distiller program version number, send a COPYDATASTRUCT containing the DM_VERSION message to Distiller. 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.

Example: ***Obtaining the Distiller 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 32-bit Windows is different from the one for 16-bit Windows since the WM_COPYDATA message and COPYDATASTRUCT structure are not defined for 16-bit Windows. Make sure you use the correct distctrl.h for your application.

Passing in command lines

In addition to the old WM_COPYDATA interface that uses the limited DISTILLRECORD structure, the newer WM_COPYDATA interface 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.

Example: ***Creating a command line string***

```
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 );
```

Example: ***Getting the Distiller version number***

To get the Distiller version number, a COPYDATASTRUCT containing the DM_VERSION message is sent to Distiller.

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

Command line options

Distiller 6.0 and above support command line and `WM_COPYDATA` interfaces and add the `--deletelog` and `--nosecurity` switches. Options can be passed as arguments on the command line.

The command line syntax is:

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

If any *`switches`* are present, they must come before any input files. Switches and input file names 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 optional commands. Either the dash (-) or slash (/) 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 a double dash (--).) 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 combine switches—give each one its own prefix (- or /).

The parameter *`inputFiles`* is a list of file names, separated by spaces or commas. Spaces and commas are both legal file name characters; if a file name contains spaces or commas, enclose it in double quotes.

To process a list of PostScript files, use this syntax:

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

The following table lists the optional command line switches.

Command line switches

Switch	Description
`--deletelog:on` `--deletelog:off`	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.

`/E` *`[pdfSettingsFilePath]`*	Opens the Distiller Adobe PDF Settings dialog box to edit the specified Adobe PDF settings file. If you specify a file name with no path, Distiller looks in its Settings folder for the file. If you omit the file name, Distiller uses the current default settings file.
	
	This switch cannot 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 box and exits when you close that window.
	
	Put quotes around Adobe PDF settings file names that contain spaces. For example:
	
	`"High Quality"`
	
	`"High Quality.joboptions"`

Switch	Description
`/F`	Enables or restricts files that can be accessed by PostScript operators, depending on the version of Distiller. Typically, this switch affects file embedding. Between Distiller 8.0 and 8.1, the sense of this switch was reversed to address security concerns. For Acrobat DC Distiller 8.1 and greater, the `/F` switch permits the PostScript file operators unlimited file access. **Caution:** Enabling unlimited file access can pose security problems. For Acrobat DC Distiller 5.0 through 8.0, the `/F` switch restricts PostScript file operators to fonts, color profiles, and other system resources needed for normal operation. Regardless of the `/F` switch, all versions of Acrobat DC Distiller can access fonts, color profiles, and other system resources needed for normal operation.

`/J [pdfSettingsFilePath]`	Uses *`pdfSettingsFilePath`* as the Adobe PDF settings file for any input files specified on the command line. If you specify a file name with no path, Distiller looks in its Settings folder for the file. If you omit the file name, Distiller uses the current default settings file. Does not affect any of the Adobe PDF settings you can set with the user interface.
	
	Put quotes around job option file names that contain spaces. For example:
	
	`"High Quality"`
	
	`"High Quality.joboptions"`

`/N`	Runs a new instance of Distiller, even if another Distiller instance 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. **Note:** When using this switch, do not use the command line `start` command with the `/wait` option. The command “`start /Wait`” negates the effect of using `/N` with Distiller.

Switch	Description
`--nosecurity`	Suppresses the Confirm Security dialog box. No security will be applied to the file. **Note:** This option was implemented only for the Adobe PDF printer and should be passed as a `WM_COPYDATA` message or through 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 user interface, then the security settings in the registry will be applied; this is the reason to show the Confirm Security dialog box.
`/O` *`outputFileOrFolderPath`*	Specifies the output PDF file name or a directory name to put PDF files in. If you specify a directory name, Distiller uses the input file name for each input file. To process a PostScript language file and name the output PDF file use this syntax: `acrodist /o` *`destFile srcFile`* To process a list of PostScript language files and place the output PDF files in a specified directory, use this syntax: `acrodist /o` *`directory srcFile1`*`[,` *`srcFile2`*`...]`
`/Q [:seconds]`	Instructs Distiller to exit immediately when it becomes idle. Distiller checks this switch only if Distiller is launched as a new instance (`/N`); otherwise idle time is limited by the watched folder timer. If `/N` is also used, Distiller quits after processing all files on the command line. If `/N` is not used, Distiller quits after processing all PostScript files in watched directories and any that were specified on the command line. To force Distiller to terminate when it has finished distilling all PostScript files in watched directories and any that were specified on the command line, use this syntax: `acrodist /q [`*`sourceFile1`*`[,` *`sourceFile2`*`...]]` `/Q` also accepts an optional timeout value in seconds, `/Q:n`. Use the timeout to instruct Distiller to wait for `n` seconds of idle time before exiting. `n` must be a positive integer and cannot be greater than 2147483 seconds (`0x7fffffff/1000`), 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 is checking every 10 seconds, it is idle for no more than 10 seconds. If the `SetTimer` value is less than 10 seconds, then the system rounds this value to 10 seconds.

Disabling prompts for output file names

Distiller prompts for output file names based on settings that can be managed with a user interface or by programmatic control.

Turning prompting off for the current user

Prompting can be turned off for the current user for all Adobe PDF document creation regardless of the application using the Adobe PDF printer.

To turn prompting off for the current user:

1. From the Windows Start menu, click **Control Panel** and open **Printers and Faxes** folder.
2. Right click on **Adobe PDF printer** and select **Printing Preferences** from the context menu. Select the default document folder from **Adobe PDF Output Folder** menu, or click the **Browse** button to create your own output folder.
3. Click **OK**.

Prompting can be turned off for the current user for Adobe PDF document creation using the Acrobat Distiller.

To turn prompting off for the current user using Distiller:

1. Start Acrobat Distiller, from the Distiller menu choose **File > Preferences**.
2. In the Preferences - Acrobat Distiller dialog box, deselect the **Ask for PDF file destination** option.
3. Click **OK**.

Customizing deployment options to turn off prompting

You can disable prompts via the Customization Wizard for Windows prior to application deployment

Programmatic control

Prompting can be turned off programmatically by adding a key to the Windows registry. This method applies to the creation of only one PDF document by a specific application for the current user. To use this method, add the following 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 is not to be used.

The *`application`* subkey is the full path of the application for which prompting is to be 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 use single slashes. 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 Apple Events

Apple events can be used from programming languages such as C or from AppleScript. Because AppleScript is more straightforward, it is recommended for use with Distiller.

Distiller supports the `application` object and the following Apple events:

- `Distill`
- `run`
- `quit`

Objects

application

application

The top-level scripting object.

Elements

`Document`, `Window`

Properties

Property	Class	Description
`postScriptVersion`	Unicode text [r/o]	PostScript interpreter version; for example, "3016.102".
`locale`	Unicode text [r/o]	Three-character language code for the Distiller user interface (for example, "ENU" is English).
`frontmost`	Boolean [r/o]	True if Distiller is the active application.
`name`	Unicode text [r/o]	Name of the application.
`version`	Unicode text [r/o]	Version of the application.

Events

Distill

quit

run

Distill

Distills a file.

Syntax

```
Distill sourcePath UnicodeText [destinationPath UnicodeText]
   [adobePDFSettingsPath UnicodeText]
```

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 is generated in the same folder as the input file.
`adobePDFSettingsPath`	Either the 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.

Returns

A Boolean value indicating status. If `true`, the command succeeded.

Examples

These examples assume the presence of an appropriate *tell – end* construct, for example:

```
tell application "Acrobat Distiller 8.0"
     [Your code]
end tell
```

Ensure each command is on one line without line breaks. Some of the following examples appear on two lines solely to fit the page.

```
Distill sourcePath "/hello.ps"

Distill sourcePath "/hello.ps" destinationPath "/Users/username/Desktop"
     adobePDFSettingsPath "Standard"

Distill sourcePath "/hello.ps" adobePDFSettingsPath "/Library/
     Application Support/Adobe PDF/Settings/"
```

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

```
set distSetting to "Standard"
```

```
if (postScriptVersion as string) is equal to "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

Terminates the Distiller program.

Syntax

```
quit
```

run

Launches the Distiller program and invokes its standard startup procedures.

Syntax

```
run
```