Adobe℠ Customer Services

Using the RunFile Procedure to Combine PostScript™ Files

The Acrobat Distiller and Acrobat Network Distiller applications support a predefined procedure called RunFile for processing multiple PostScript files into one Portable Document Format (PDF) file. This technical note describes how to run this procedure using a PostScript file created with a text editor. This is a convenient alternative to combining multiple PDF files to create a single PDF file using the Adobe Acrobat Exchange insert command.

CREATING A POSTSCRIPT FILE TO THE RUNFILE PROCEDURE

Note: Be sure to use the exact syntax used in this document, including spaces and capital letters.

1 Using a text editor, create a new file.

2 Enter the following:

```
%!
/prun {/mysave save def dup = flush RunFile clear cleardictstack
mysave restore} def
```

The first line is a comment that identifies this file as a PostScript file. The next line defines a procedure to access the RunFile procedure.

3 Enter the pathname of a PostScript file to process, in parentheses, followed by prun. Use the following syntax

• For the Windows version of Distiller:

```
(C:\\REPORTS\\JANUARY\\JANSALES.PS) prun
```

• For the Macintosh version of Distiller:

```
(Hard Disk 80:Reports:January:January Sales.ps) prun
```

This instructs Distiller to process the file JANSALES.PS in the directory JANUARY within the directory REPORTS on drive C:.

In this case, the Distiller would process the file January Sales.ps in the folder January within the folder Reports on the disk drive named Hard Disk 80.

Pathnames are always relative to the computer running the Distiller, not necessarily the machine that created the file. This is especially important for users of the Acrobat Network Distiller program, because the pathname to an In directory or folder used by the Network Distiller may be very different from the pathname to the same directory or folder used by a local workstation. This information can be obtained from your system administrator.

4 Repeat step 3 as necessary for each PostScript file you want to process.

5 Designate the end of the file by entering:

```
%EOF
```

6 Save your file as a text only file. Verify that all PostScript files are in the locations specified by the pathnames you entered.

7 Place the file in the Distiller's In directory for processing. When you use the Distiller to process this file, it will read sequentially all PostScript files specified and output one PDF file with the same name as your file but followed by a .pdf extension. The name of each PostScript file will appear in the Distiller's Messages window as it is being processed and appear in a log file that will be created regardless of whether an error has occurred.

Here is a complete example of using the RunFile procedure to combine three PostScript files into one PDF file using the Windows version of Distiller:

```
%!
/prun {/mysave save def dup = flush RunFile clear cleardictstack
mysave restore} def
(C:\\WORK\\TITLEPG.PS) prun
(C:\\WORK\\BODY.PS) prun
(C:\\WORK\\INDEX.PS) prun
%EOF
```

If this file is saved as FINAL.PS and opened with Distiller, the three specified PostScript files in the directory WORK on drive C: of the machine running Distiller would be processed and a file named FINAL.PDF would be generated.

USING THE RUNFILE PROCEDURE WITHOUT CREATING A LOG FILE

If you do not want the Distiller to display the PostScript filenames in the Messages window nor create a log file unless an error has occurred, follow the directions above, except in the second line of step 2 substitute the following:

```
/prun {/mysave save def RunFile clear cleardictstack mysave restore}
def
```

USING THE RUNFILE PROCEDURE WITHOUT CLEARING THE DICTIONARY STACK

If you include PostScript files that are not completely self-contained (i.e., they rely on a prolog having been run that leaves a procset dictionary on the stack), you must use a different procedure; otherwise errors may occur during the distilling process and the resulting PDF file may be incorrect. However, by not clearing the dictionary stack, you increase the probability that the Distiller will run out of memory and return with an error that would not have occurred had the PostScript files been distilled separately. Increasing the memory available to the Distiller may correct this problem.

To use the RunFile procedure without clearing the dictionary stack, follow the directions above, except in the second line of step 2 substitute the following:

```
/prun {dup = flush RunFile} def
```

If you also do not want a log file created unless an error has occurred, you can omit the second line of step 2 and access the RunFile procedure directly by substituting RunFile for prun in step 3. For example:

```
(C:\\REPORTS\\JANUARY\\JANSALES.PS) RunFile
```