PDF ePrinter

Reference Manual

For Version 1.5 ***May 27, 2001***

SOFTWARE LICENSE AGREEMENT

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE EXECUTING PDF EPRINTER. EXECUTING PDF EPRINTER INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THEM, DELETE PDF EPRINTER FROM YOUR COMPUTER AND ALL ASSOCIATED STORAGE MEDIA.

0. Introduction
PDF ePrinter is distributed as freeware. Since the meaning of this term varies within the software community, this license serves to define the terms under which PDF ePrinter may be used and redistributed.

1. Definitions and Applicability
This license applies to PDF ePrinter ("the Software") as distributed by Matthew Dinmore ("the Author") to include all components of the original distribution package.

The end user ("You") is legally bound by the terms of this license if you use, redistribute or incorporate the Software in a larger work. By using the Software, you acknowledge acceptance of the terms of this license and are bound by them. If you do not accept the terms of this license, you must discontinue using the Software and delete it from your computer and any associated storage media.

2. License
This license permits you to install and use the Software on as many machines as you like without paying the Author a fee. You may redistribute the Software in its entirety in accordance with the provisions set forth in Section 5 of this license. You may use the Software in its entirety as a component of a larger work. You may not disassemble, decompile or reverse engineer the Software.

3. Disclaimer of Warranty
THE SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, THAT THE SOFTWARE IS FREE FROM DEFECTS, AND FITNESS FOR A PARTICULAR PURPOSE.

4. Limitation of Liabilities
THE ENTIRE RISK OF USING THE SOFTWARE IS ASSUMED BY THE USER. IN NO EVENT WILL THE AUTHOR BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL LOSSES OR DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE THE SOFTWARE.

5. Redistribution
The software may be redistributed in accordance with the following restrictions:

(a) The Software must be distributed in its entirety as originally distributed by the Author. No portion of the software may be removed or modified, and no additional items may be added to the software package.

(b) You may redistribute the Software, but you may not charge a fee for it. You may charge a nominal media/distribution fee. The following is a list of some common redistribution scenarios allowed by this license (this is not exhaustive):

i. The Software may be redistributed by on-line services which charge a service access fee, so long as this fee is not specific to accessing and downloading the Software.

ii. The Software may be redistributed by users groups. If the users group charges a membership fee, the fee may not be specific to obtaining the Software.

iii. The Software may be redistributed as part of a larger collection by a commercial publisher, to include as a complement to a print publication, for a nominal media and distribution fee.

iv. The Software may be used as a component of a "custom software system" provided by a systems integrator for which a consulting/development fee is received. HOWEVER, THE SYSTEMS INTEGRATOR MUST

CLEARLY REPRESENT TO THE END USER THAT THE SOFTWARE IS FREELY AVAILABLE TO THEM EXTERNAL TO THEIR CONTRACT WITH THE SYSTEMS INTEGRATOR, AND THAT THE SYSTEMS INTEGRATOR HAS NOT CHARGED THEM FOR THE SOFTWARE ITSELF. THE SYSTEMS INTEGRATOR MAY NOT USE THE SOFTWARE IN THE PROMOTION OF THEIR SERVICES OR PRODUCTS. THE SYSTEMS INTEGRATOR MUST INCLUDE THE AUTHOR'S COPYRIGHT NOTICE ON THE LARGER WORK.

6. Promotion

The Author's name and the Software may not be used in any promotion, advertisement or endorsement, either directly or indirectly, without expressed prior written consent of the Author.

7. Termination

This license will be terminated automatically if you fail to comply completely with its terms, and fail to correct any non-compliance within 30 days of becoming aware or being notified of a breach. Upon termination, you must discontinue using the Software and destroy all copies of it in your possession. If you distribute the Software or incorporate the Software in a larger work, you must immediately cease distribution or use of the Software in your custom software solution. End users who received the Software from you will not be liable for you r breach of the license, so long as they are in compliance with the terms of the license, they may continue using the Software.

8. General

This license represents the complete agreement between you and the Author with respect to the Software. This agreement is governed by the laws of the United States of America and the State of New York. If any provision of this license is held to be unenforceable in a legally recognized court of law in the United States of America, that provision will be enforced to the maximum extent possible, and all other provisions will remain in full force and effect. Previous agreements, if any, are hereby expressly cancelled. This agreement may not be changed or amended except in writing by the Author.

The End

Contents

Overview

PDF ePrinter is an application which converts print jobs generated by the Apple LaserWriter® or StyleWriter® printer drivers into Portable Document Format (PDF) files. It allows you to customize the conversion process and take advantage of PDF's features for creating outlines, making hyperlinks and controlling the presentation of the document. Customized setting can be saved as a template and reused for future jobs. PDF ePrinter also supports AppleScript, allowing jobs to be customized and generated without user intervention, enabling PDF ePrinter to be used as part of a production workflow.

This manual describes how to install and use PDF ePrinter. It also provides technical documentation about scripting with PDF ePrinter. For more information on the PDF file standard, see the current version of the Portable Document Format Reference Manual, which can be found at:

http://partners.adobe.com/asn/developer/acrosdk/DOCS/PDFRef.pdf

How PDF ePrinter Works

PDF ePrinter is not a printer driver. Instead, it uses the standard Apple LaserWriter (version 8.5.1 or newer) or StyleWriter 1200 driver to create a spool file, which PDF ePrinter then converts to a PDF document. It uses a custom desktop printer created by the Desktop Printer Manager to "hold" the file until it is processed by the application. This approach takes advantage of the drivers' printing features[1], as well as the Desktop Printer system, without introducing a new driver.

Print jobs can be processed immediately if the PDF ePrinter application is running; it will "watch" the desktop printer for new jobs and handle them as they arrive. Alternately, jobs can be held in the queue and processed later.

About ClibPDF

PDF ePrinter uses a slightly modified version of ClibPDF, a library of functions which provide an application programming interface for generating PDF files. ClibPDF was written by FASTIO Systems, Inc. and is available at their website at http://www.fastio.com/ . Since all PDF generation is performed through this library, PDF ePrinter takes advantage of many of its special abilities, but is also restricted by the practical limitations of the library. For most general applications, these will not cause any problems. However, an advanced user may wish to look at the ClibPDF reference manual to learn more about the library's capabilities. Some of the advanced features made available through PDF ePrinter are not documented in detail in this manual, but are fully explained in the library's documentation.

[1] Only those features which are not specific to its generation of PostScript.

Installation

There are several ways to install PDF ePrinter: use the ePrinter Setup Utility to create an ePrinter desktop printer, or create it yourself using the Apple Desktop Printer Utility or Chooser (provided with the system software). You must have at least LaserWriter 8.5.1 and/or StyleWriter 1200 and the Desktop Printing software installed and active. These items are located in the Extensions folder of the System folder. If you do not have these, install them first following Apple's installation instructions (note: you do not need to have an actual LaserWriter or StyleWriter printer in order to install these drivers).

Which Driver Should I Use?

PDF ePrinter v1.5 offers the choice of three different drivers: LaserWriter 8, StyleWriter 1200 or Color SW 1500[2]. You can install either or both and switch between them.

The LaserWriter driver is the best choice for most applications. PDF ePrinter is optimized to process spool files generated by the LaserWriter, and will take advantage of features such as rotated text and graphics. However, some applications recognize the fact that they are printing to the LaserWriter driver and generate PostScript® Language files embedded within the QuickDraw® spool file. PDF ePrinter cannot process PostScript Language and will ignore it (you will be warned that the file contains PostScript Language). Some applications only generate a portion of their output as PostScript Language, in which case only some of the output may be missing (or possibly not at all if the application has included the equivalent QuickDraw commands). Other applications generate their entire output as PostScript Language; the result will be a series of blank pages in the PDF file.

The StyleWriter driver avoids this problem because applications will not try to produce PostScript Language when printing to it. The StyleWriter has its own limitations, namely in how it handles bitmapped images. You cannot use the Faster picture rendering option, QuickTime™ pictures will not appear, and thumbnail representations may be inaccurate when using the StyleWriter driver. Additionally, some applications will not generate rotated text or graphics but instead create a bitmapped representation which will result in a lower-quality and larger-sized PDF file. You should use this driver only if the LaserWriter produces unacceptable results. Note that some high-end graphics applications *only* print to PostScript Language printers; they may not print to the StyleWriter at all. These applications are incompatible with PDF ePrinter.

You can try other drivers, as well. See the section Other Printer Drivers below.

Non-Roman Language Systems

If you are using a non-Roman language system, you can use PDF ePrinter's ability to generate text as graphics. Although less-than-ideal, it will work. For Chinese, Japanese and Korean fonts, several built-in substitution fonts are available. Using these will require the reader to have the Asian Fonts Pack installed.

---

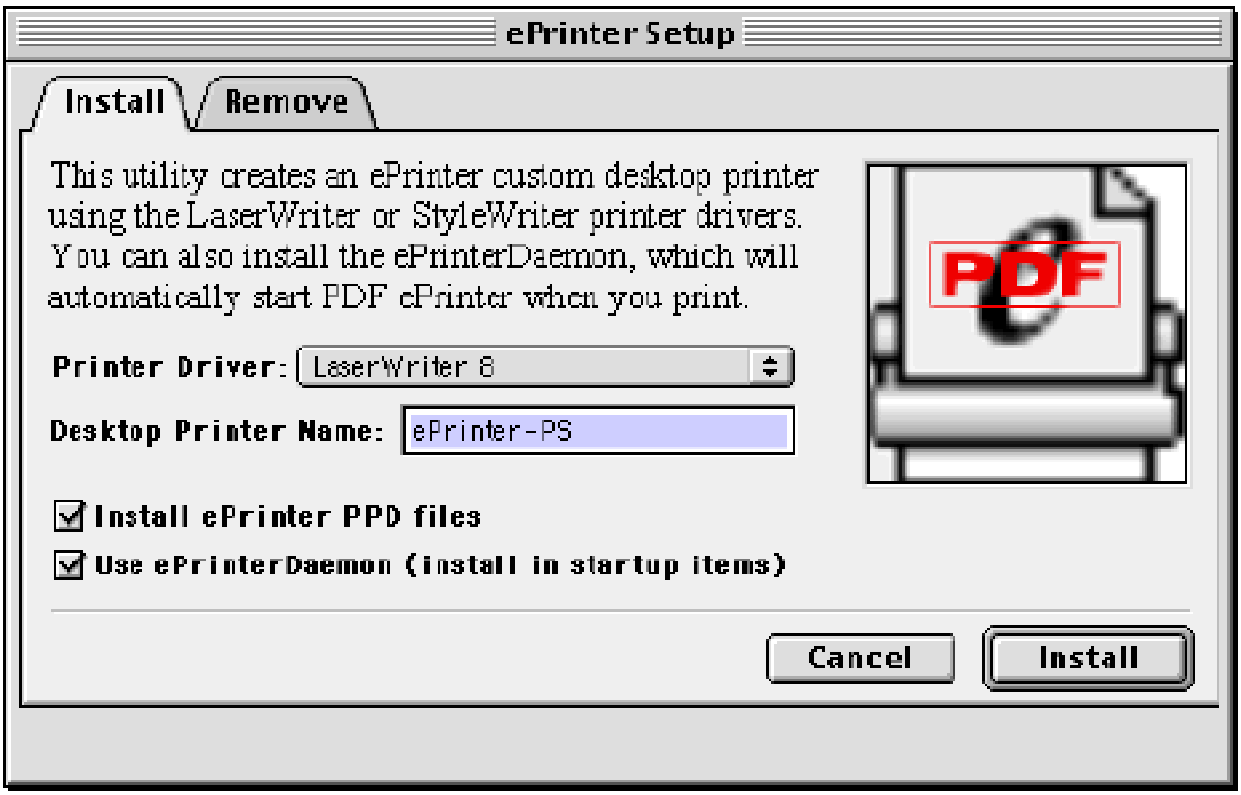
[2]Other StyleWriter-family drivers may work, but have not been tested. There does not appear to be any difference in the capture phase with these drivers, so they should all work equally well with PDF ePrinter.

Important: In order to print non-Roman characters, you must use the StyleWriter driver. The LaserWriter automatically converts these characters to graphics in a manner PDF ePrinter cannot render to PDF.

Using the ePrinter Setup Utility

This utility is built into PDF ePrinter; it will start automatically the first time you run PDF ePrinter, and you can access it at any other time by choosing ePrinter Setup from the File menu to access it.

The Install tab allows you to choose which driver (only LaserWriter is supported from this dialog at this time; use the Chooser to install StyleWriter 1200) and set the name of the desktop printer that will be created. You can also install the ePrinter PPD files (necessary only for the LaserWriter driver), and set up the ePrinterDaemon. Click the Install button to create an ePrinter desktop printer (note that the necessary drivers and desktop printer software must already be loaded. If it is not, you will be notified of this and you should load the software from your system software disk).



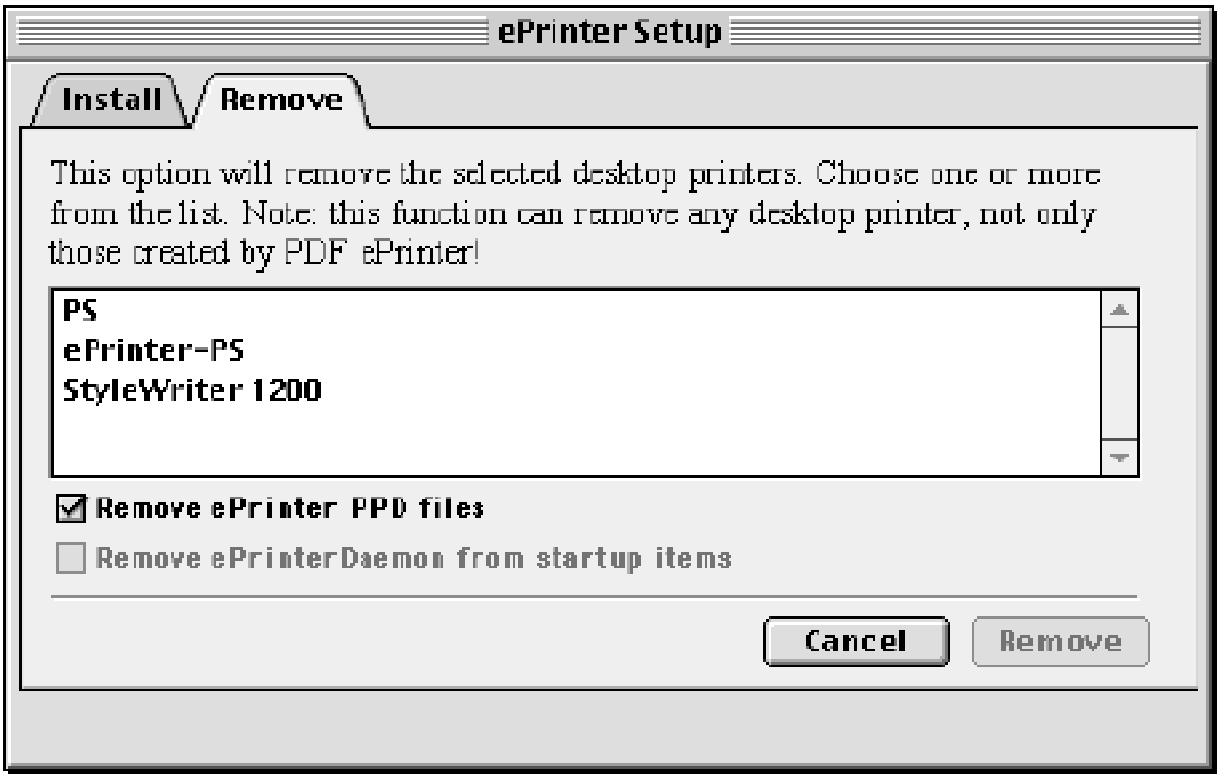
The ePrinter Setup Install Tab

The installation option, 'Install ePrinter PPD files,' should be left checked the first time you install ePrinter. The only time you might want to uncheck this is of you have modified the ePrinter PPD file and don't want your changed file to be overwritten if you reinstall ePrinter.

You also have the option of installing the ePrinterDaemon. This is a very small application that runs in the background and watches the ePrinter desktop printer folder for new jobs. When one

appears, it will launch PDF ePrinter to process the job, and then (optionally) shut it down again when the job is completed. This not a system extension and the only thing installed by this option is an alias to the ePrinterDaemon application.

When you click the Install button, the ePrinter desktop printer will be created and the other options processed. If there are any problems, you will be notified, and you may have to manually create the desktop printer with the Apple Desktop Printer Utility, as described below. Note: PDF ePrinter uses the Desktop Printer Manager scripting addition to create and manage desktop printers, so you must have this loaded (you should unless you have modified your system). Also, there is a problem with this addition that causes the icon for the desktop printer to not be properly set initially. When you next restart your computer, it will have the correct appearance.

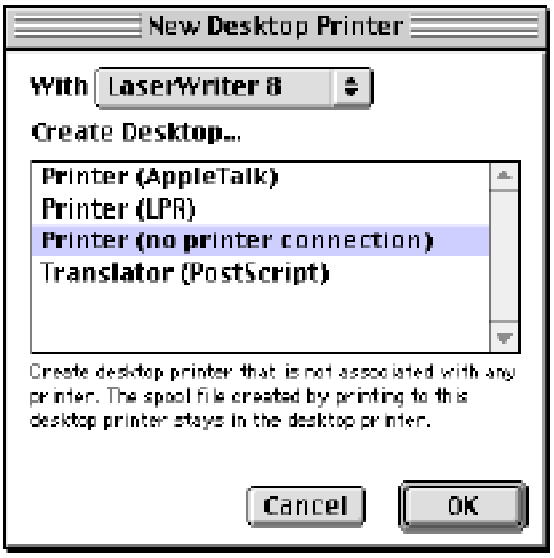


The ePrinter Setup Remove Tab

When you use the utility's Remove feature, you can select one or more desktop printers to remove. **Note that all desktop printers are listed here, not just those created by or for PDF ePrinter.** You can also remove the PPD files and the alias to ePrinterDaemon, if you installed it. When you delete a desktop printer, if there are still jobs in the printer's queue, you will get an error message from the Finder.

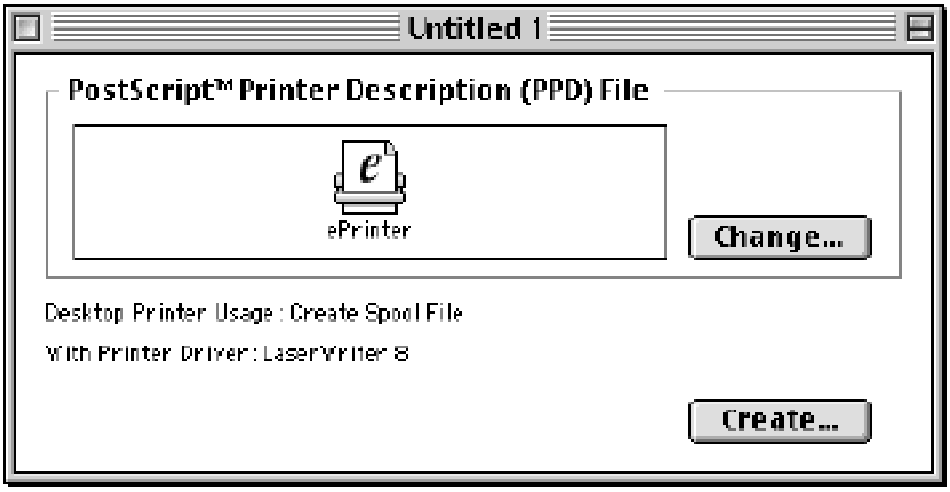
Using the Apple Desktop Printer Utility

This application is installed with the LaserWriter software; it can typically be found on your start-up hard disk within the Apple Extras:Apple LaserWriter Software folder.



The Apple Desktop Printer Utility

Run the utility. A window will appear. From the list, select "Printer (no printer connection)." Then click OK.



Desktop Printer Utility PPD Configuration Window

A new window will open allowing you to select a PPD file. You can use the Generic PPD file (the default), or the ePrinter PPD file (to use this, you must copy this file to the Printer Descriptions folder within the Extensions folder; it can be found in the Components folder of the ePrinter installation. Make sure to leave a copy in the Components folder for PDF ePrinter to use). Then click the Create... button. This will display a standard file dialog; you should name the new printer "ePrinter" and place it on the desktop (if you do not, and you want PDF ePrinter to automatically process spool files, you will need to change the location of the watch folder in PDF ePrinter's Preferences dialog). Then click the Save button.

Using the Chooser

You can also use the Chooser. Doing so will create a desktop printer you can use with PDF ePrinter. This is the recommended way to install the StyleWriter 1200 or Color SW 1500 desktop printer.

Open the Chooser and select the appropriate StyleWriter icon. In the listbox on the right, select one of the ports (it doesn't matter which one). If you don't have any physical serial ports or cannot select any, you will still be able to create the desktop printer; just select "<no printer port>." After closing the Chooser, when the desktop printer icon appears, be sure to set its print queue to Pause (a stop sign badge should appear on the icon). To do this, select the printer on the desktop and choose "Stop Print Queue" from the Print menu. You should then select this printer in the PDF ePrinter Preferences dialog.

Note: the appearance of the red "Stop Badge" on the printer is normal. This is because the printer queue is paused which is necessary for PDF ePrinter to be able to capture the print job before the system's print manager tries to print it to an actual printer. If you install the StyleWriter 1200 from the Chooser, it is important that you set the resulting desktop printer's queue to Paused.

Using PDF ePrinter

PDF ePrinter was designed as a semi-automated, post-processing application. This means that it will typically be used for specific jobs where PDF output is required from an application. As such, it is only necessary to run PDF ePrinter when you intend to produce PDF files. However, you may choose to automatically start it and leave it in the background until it is required, or you may start it automatically as it is needed via an AppleScript or other mechanism.

The general procedure for using PDF ePrinter is as follows:

1. Prepare a document for conversion to PDF by opening it in its creating application and preparing it for printing. Make sure you use Page Setup to choose the correct paper size. You can create custom paper sizes by modifying the ePrinter PPD file.
2. Set the default printer to the ePrinter desktop printer (by selecting its icon on the desktop and choosing Set Default Printer from the Printing menu or Control-clicking its icon and choosing Select as Default Printer from the contextual menu).
3. For immediate post-printing conversion to PDF, start PDF ePrinter. Alternately, you can print several documents which will be stored in the desktop printer's queue and processed the next time you run PDF ePrinter.
4. Print the document from its creating application.
5. If PDF ePrinter is running, it will open a job window for the document (by default, this will come to the front, although you can change this behavior in the Preferences dialog).

If a default template has been selected, it will be loaded automatically.

6. Make whatever processing settings changes are necessary, and click the Create PDF button.

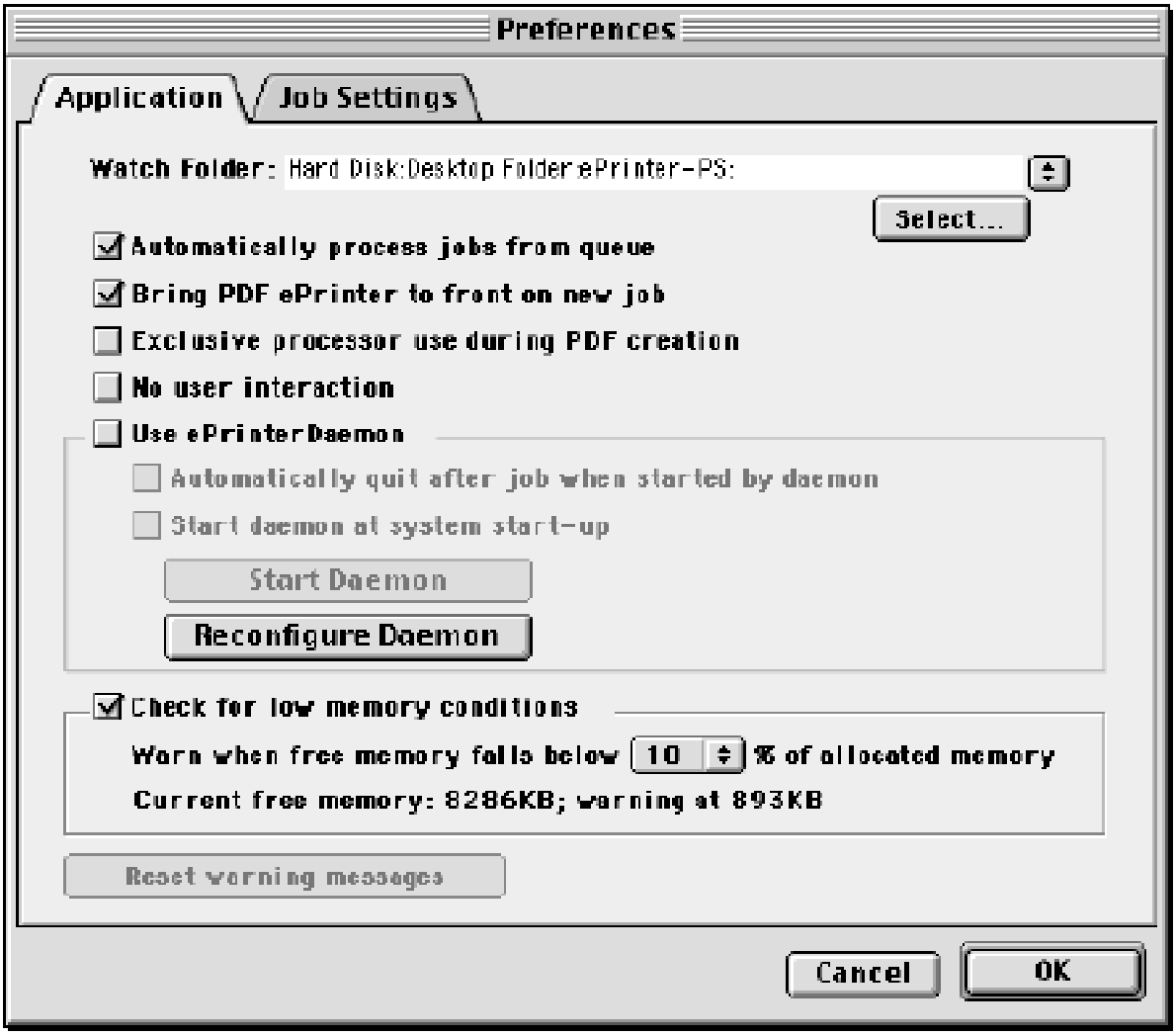
7. Unless you have set a default target folder, you will be asked to name the output PDF file.

8. The PDF file will be created and the spool file removed from the queue.

After PDF ePrinter has processed the job, it will close the job window. If there is another job in the queue, it will open a new job window. When you are done, you can either Quit PDF ePrinter, or leave it open in the background.

Configuring PDF ePrinter

You can set several general configuration options for PDF ePrinter by running it and selecting Preferences... from its Edit menu; the following dialog will appear:



PDF ePrinter Preferences Dialog - Application Tab

Watch Folder: the path to the folder PDF ePrinter will continually check for new spool files. Normally, this points to the desktop printer named "ePrinter-XX" where XX is PS or QD depending on the type of printer. You can also choose any regular folder by clicking the Select... button or by typing a full pathname; be sure to include the final colon (:). You can use the drop-down menu arrows on the right to select any desktop printer (only printers actually located on the desktop are displayed). Note that although some other types of printers *may* work with ePrinter, they may need to be specially configured; for more information on trying other drivers, see Other Printer Drivers.

Automatically process jobs from queue: Normally, this will be checked to cause PDF ePrinter to continually check the selected printer queue and open jobs as they are printed. You might want to uncheck this option if you plan to control PDF ePrinter from an external script.

Bring PDF ePrinter to front on new job: when checked, the PDF ePrinter application will come to the front of all other applications when it detects a new job in the watch folder. If unchecked, it will still open a dialog for the job, but it will remain in the back.

Exclusive Processor use during PDF creation: when checked, PDF ePrinter will "monopolize" the computer's processor - no other applications will be able to run and you will not be able to interact with the interface. This may provide improved performance, particularly for longer jobs. By default, this is disabled and you will be able to use other applications while PDF ePrinter is working in the background.

No user interaction: when selected, PDF ePrinter will not display user dialogs during PDF creation. Warnings will be handled as if the user pressed "Cancel," and font substitutions will be automatically set to Times-Roman. This option is useful for unattended operations such as a server-based application.

Use ePrinter Daemon: when checked, the daemon is started. This small application watches the selected watch folder when PDF ePrinter is not running and launched PDF ePrinter when a new job appears. If the next option, *Automatically quit after job when started by daemon*, is checked, PDF ePrinter will quit when a job initiated by the daemon is completed. You can choose to start the daemon automatically when the computer starts by checking the *Start daemon at system start-up* option.

The *Start Daemon/Stop Daemon* button can be used to control the daemon. If its label is "Start Daemon," then the daemon is not running and clicking the button will start it. Conversely, if the label reads "Stop Daemon," then the daemon is running and clicking the button will stop it. Note that these actions are immediate; you do not need to click the OK button at the bottom of the window. Also, a message may appear next to the Start/Stop button telling you what will happen when the OK button is clicked, depending on the current settings.

The *Reconfigure Daemon* button will reset the daemon's watch folder and target application internal parameters to the proper values. You may need to do this if you move the PDF ePrinter folder or otherwise changes the names in either of these paths (if either of these happens, you

will get an error when the daemon tries to start, telling you to reconfigure it from your ePrinter application).

Check for low memory conditions: when enabled, watches the allocated heap and warns you if it falls below some percentage of the original size (10%, by default). If you increase the amount of memory to PDF ePrinter in the Finder, you may want to lower the percentage here. A rule-of-thumb is that you should have at least a 250KB margin. When PDF ePrinter runs with its default memory settings, it has a free heap of about 2.5MB, hence the default 10% setting. See the Troubleshooting section for more information on memory allocation. The current free memory and warning point are also displayed. The warning point is based on the amount of memory available when PDF ePrinter started; this may decrease over time and will not necessarily be related to the current free memory value.

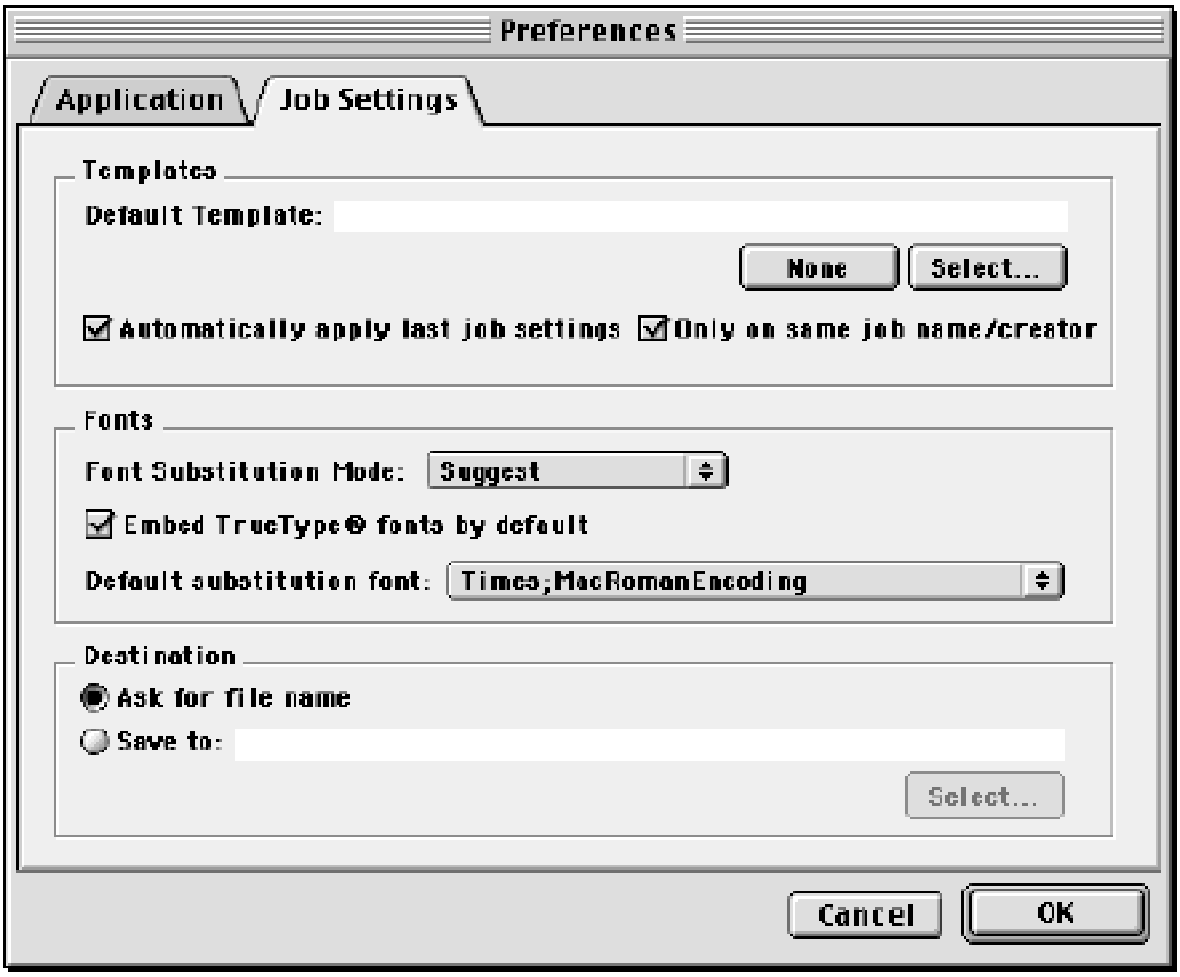
Reset Warning Message button: when enabled, there are warning messages you have requested not be displayed. Clicking this button will reset these messages so they appear again. The warning messages reset by this command are:

PostScript warning: displayed when PostScript is encountered in a job.

ePrinter not Installed: displayed when the default desktop printer cannot be found.

Job Retry: displayed when a job fails to complete asking whether you want to retry it with new settings.

Faster Picture Rendering warning: displayed when Faster Picture Rendering is on and a StyleWriter printer driver was used to generate the job. Note that this is automatically turned on for some applications; if this is the case, you should try to use the LaserWriter driver to generate the document with those applications.



PDF ePrinter Preferences Dialog - Job Settings Tab
Default Template: choose a template file (created with the Save Template command) that will be automatically applied to all new jobs. Click the None button to disable this feature.

Automatically apply last job settings: when selected, will restore all the settings from the previous job (these are stored automatically for each job as a special template file). The associated option, *'Only on same job name/creator*,' will only apply the settings if the name and creating application of the print job is the same.

Font Substitution Mode: The method PDF ePrinter will use to handle unknown fonts during a job. Options are:

> *Manual*: You will be prompted when PDF ePrinter encounters a font it doesn't know how to handle. You will have to choose the best way to handle the font.
>
> *Suggest*: PDF ePrinter will suggest a font to use in place of the unknown font. This may include preparing a TrueType font for use. If you don't accept the suggestion, you must manually select a way to handle the font.
>
> *Automatic*: PDF ePrinter will make the font substitution without prompting you.

Embed TrueType® fonts by default: When preparing a TrueType font, the automatic behavior will be to embed the font if this is set. This affects both the Automatic and Suggest font substitution modes, as well as the Font Assistant's default setting for TrueType fonts.

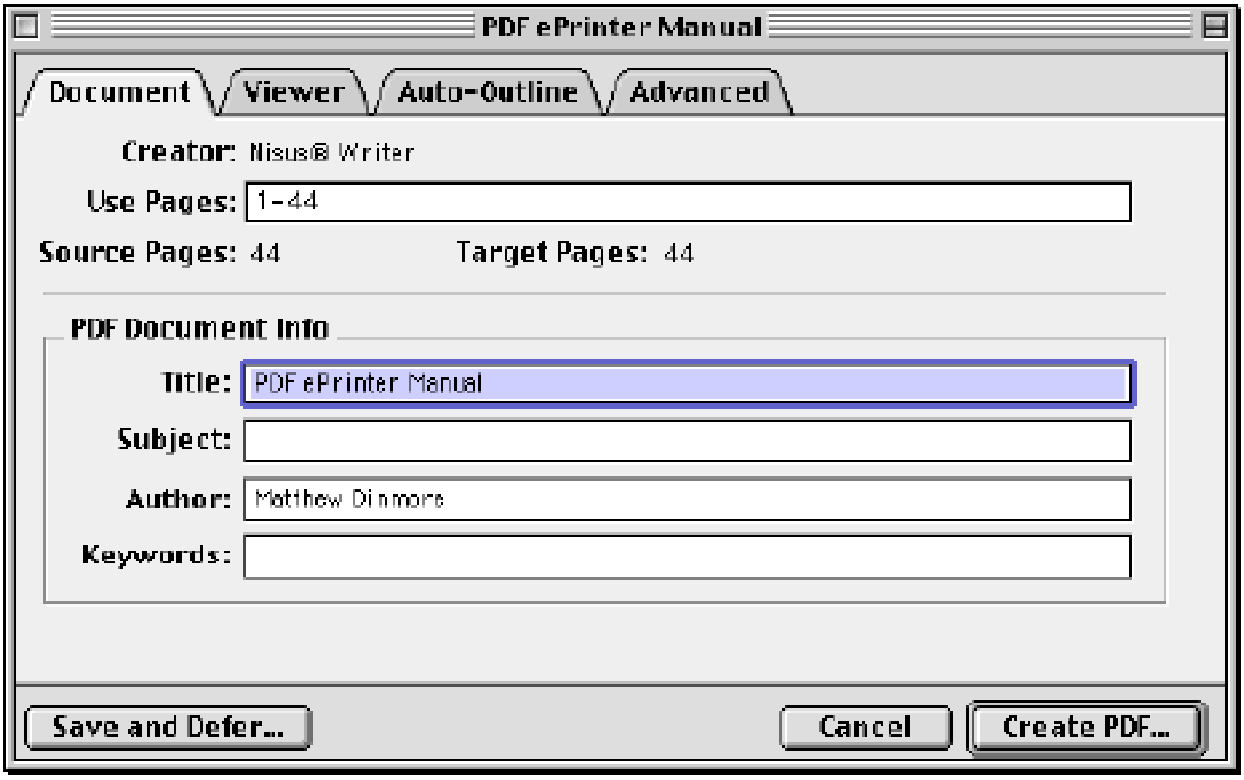
Default substitution font: the name of the font and its encoding to use no other font can be found to use for an unknown font.

Destination: Either select Ask for file name to cause PDF ePrinter to request a name (using the standard Save As dialog box) for each PDF file it creates, or select the Save to: option and choose a destination folder. With this option, the file is automatically named based on the title supplied in the job settings (with .pdf appended to the end) and saved to the selected folder.

Click OK to save the settings and Cancel to keep the previous settings. Note: settings are stored in the PDF ePrinter Prefs file within the Preferences folder. To reset all preference to the defaults, delete this file (while PDF ePrinter is *not* running) and restart the application.

Job Settings

Each time you print from an application to the ePrinter desktop printer, and PDF ePrinter is active, a job window will open. This window will allow you to set parameters which define how PDF ePrinter will convert the print job into a PDF file. These parameters are presented in a dialog with four tabs: Document, Viewer, Auto-Outline and Advanced. All of these settings can be saved and restored as a template file, or programmatically controlled through AppleScript. Additionally, some settings can be set automatically for each job based on the printer or application used to create the job.



The Job Window with the Document Tab selected

Document Tab

This tab contains general information that is stored in the PDF file's Document Info dictionary (accessible by selecting Document Info... General... from Acrobat Reader's File menu). The fields are:

Creator: the name of the application which created the document. In this case, the application from which you printed. In Acrobat Reader, the name will have the text "via PDF ePrinter v1.X.X" (depending on the version) appended. You cannot edit this field.

Source Pages: the number of pages in the document as originally printed.

Use Pages: lets you specify which pages or page ranges to use in the output file. By default, the entire input file is represented by the range '1-n' where 'n' is the total number of source pages. You can enter a string which includes individual pages and page ranges separated by commas. Page ranges must be from a lower page number to a higher page number, and the two numbers are separated by a hyphen. You can also specify a page number relative to the end of the document by preceding the page number with a dollar sign ($). An offset of zero, represented by '$0', is the last page of the source file, '$1' is the second-to-last page, etc. You can also specify that a page may be used more than once. Example: 1-4,6,8,10-$1: includes the first four pages, pages 6 and 8, and pages 10 through the second-to-last page.

Target Pages: this field shows the total number of pages that will be in the output file based on the specification in the Use Pages field. This field recalculates automatically anytime the data in

the Use pages field changes. It will also display a message if the Use Pages specification contains an error or a page that is out of range.

Title: the name of the file or document as supplied by the printer driver. This can be edited.

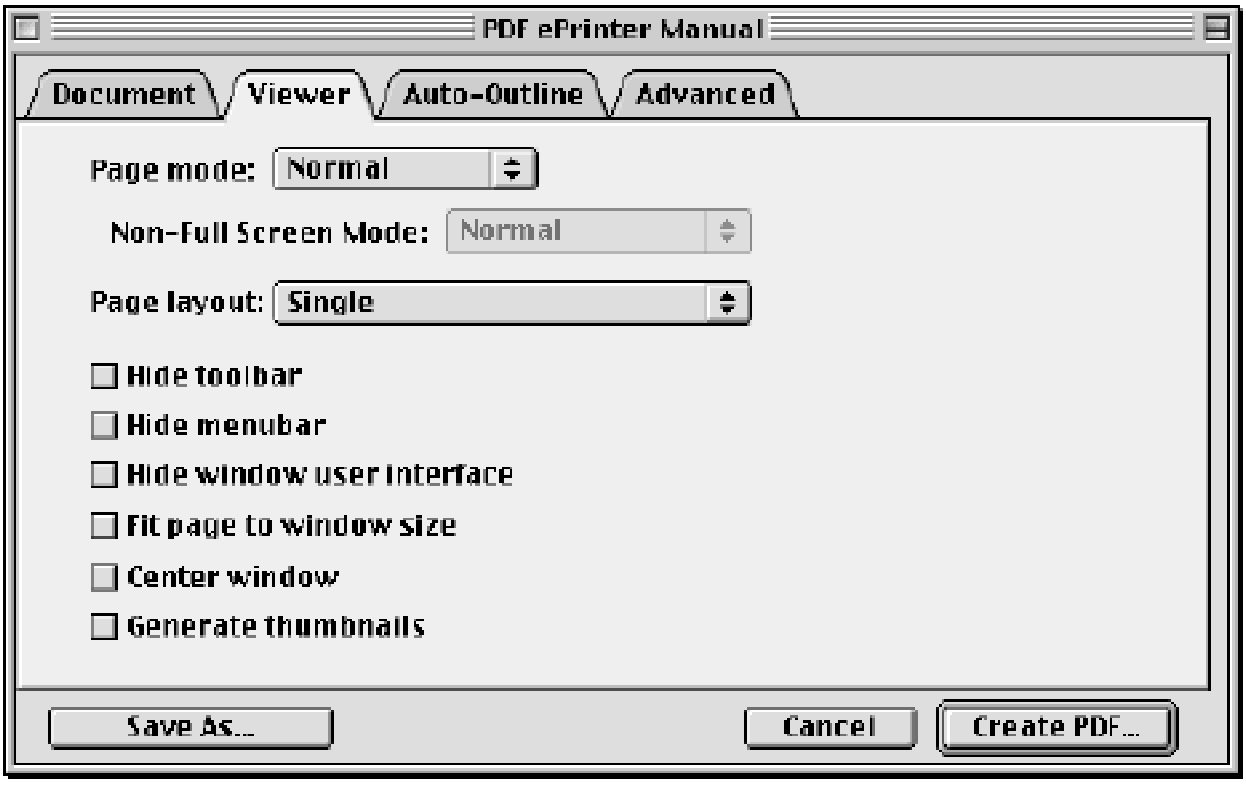
Subject: you can provide text here to summarize the document's subject matter.

Author: the name of the person who wrote the document. Be default, this will be set to the computer's owner's name if none is provided by an automatically loaded template.

Keywords: some applications which process PDF files may use this information for indexing.

Viewer Tab

The parameters on this tab define how the document will be displayed by the viewer (typically Acrobat Reader; however, these setting are general to PDF and may or may not be implemented in a particular viewer). The parameters are:



The Job Window with the Viewer Tab selected

Page Mode: determines whether the viewer opens with just the document visible (Normal), with the outlines tree opened (Outlines), with the thumbnails view opened (Thumbs), or in fullscreen/presentation mode (Fullscreen). If you select Fullscreen, the next menu becomes available allowing you to select which of the other non-fullscreen modes you want the document to appear in if the user exits fullscreen mode.

Page Layout: indicates whether a single page is displayed (Single), a continuous column of pages is displayed (1 Column), or if the pages are displayed in a double column with facing pages next to each other (either 2 Column, odd pages left or odd pages right).

Hide toolbar: when checked, the viewer's toolbar is not displayed.

Hide menubar: when checked, the menubar at the top of the screen (or window) is hidden.

Hide window user interface: hides the various controls that the viewer may display in the document window.

Fit page to window size: scales the displayed page to fit the size of the viewer window.

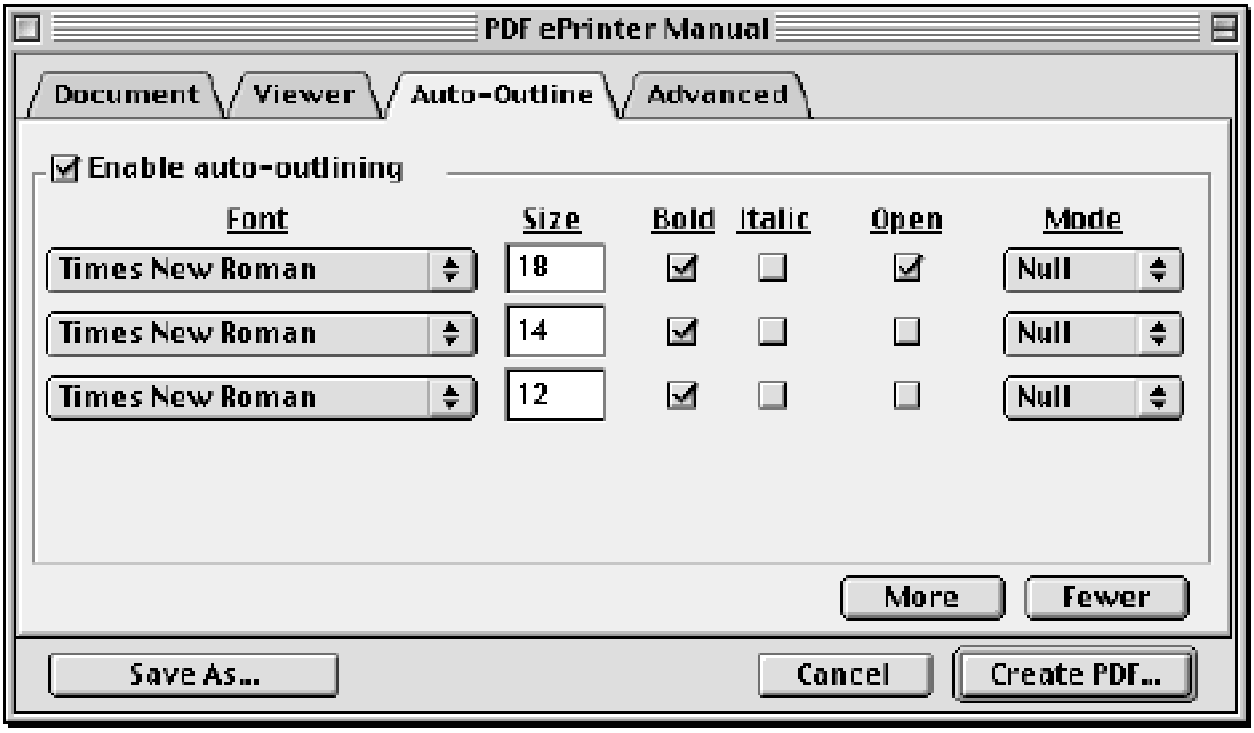
Center window: centers the document window on the display.

Generate thumbnails: creates miniature images of the document's pages that can be viewed in the PDF reader's thumbnails view. Thumbnails are created after the PDF is processed. They may vary slightly from the PDF because they are generated from the original PICT image of the page. Effects such as reversed or rotated text may not appear correctly or at all in the thumbnail. Additionally, meta-commands embedded in the text *will* appear in the thumbnails.

Note: as of Acrobat® Reader™ 5, thumbnails are automatically available in the viewer even if they were not generated as part of the PDF file. You may still want to include them for compatibility with older versions of Acrobat Reader and other applications.

Auto-Outline Tab

Auto-outlining allows an outline tree to be automatically generated for a document. This is done by defining outline styles that can be detected within the document that indicate where an outline entry should be made. For example, in this document, major sections have headings which are in Times New Roman, 18pt, bold type, and minor section heading in Times New Roman, 14pt, bold type. By defining auto-outline styles for each of these and enabling auto-outlining, PDF ePrinter will build an outline tree which contains the text of each of the headings in one of these styles at the proper level of indenture.



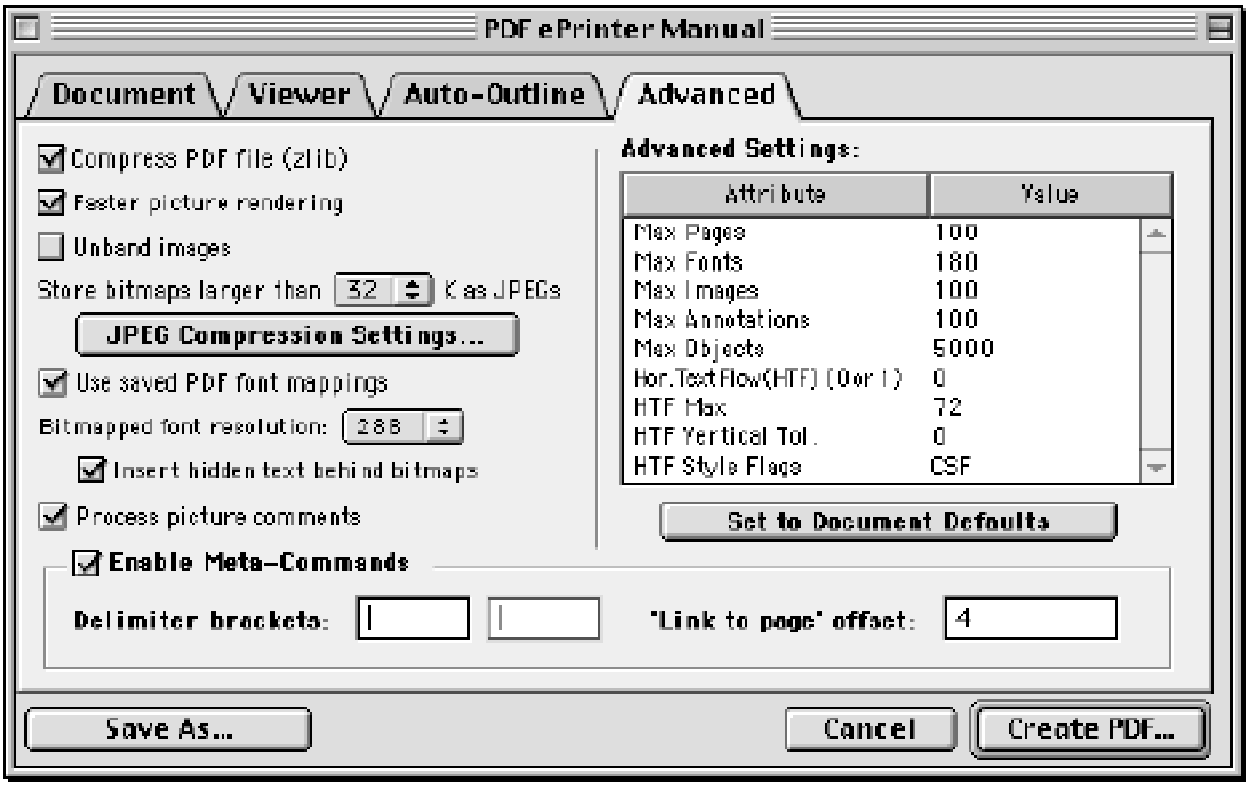
The Job Window with the Auto-Outline Tab selected

The Auto-outline tab allows you to define up to five styles that are combinations of a particular font, size and face (bold or italic). For each style, you can select whether it is initially open (i.e. all of it sub-entries are displayed when the document is opened) or closed, and the display mode used when the user selects the entry from the outline tree. The modes are Null (the new page is displayed the same way as the current page), fit (the new page is scaled to fit the entire page in the window), fitH (the new page is scaled to fit horizontally in the window), fitV (the new page is scaled to fit vertically in the window), and Y (the page is not rescaled, but is scrolled so that the outline entry is visible, based on its Y coordinate). The styles are in order of prominence with the first being the highest level and the fifth being the lowest level. Click the More button to add another level and the Fewer button to remove the last displayed level.

The Enable auto-outlining checkbox must be checked in order for PDF ePrinter to actually process any of the styles, even if you have defined styles.

Advanced Tab

The parameters on the Advanced tab allow fine control over the creation of the PDF file. The default settings are normally sufficient for most jobs, however you may have special requirements that require some tweaking of these values. Be careful when making changes; improper settings here can lead to problems with the output or even crashes!



The Job Window with the Advanced Tab selected

Advanced Settings

The upper-right side of the tab contains a listbox with several attributes and their values. These are used by ClibPDF to allocate the necessary structures for creating the PDF file, as well as managing other fine control settings for the production of the PDF file.

The first five are the memory allocation settings. By default, these are based on a maximum of a 100-page file. PDF ePrinter will automatically scale these settings for larger files. However, if you are having problems with memory, or have a document that has many fonts or images, you may want to change these values independently.

Max Pages: the total number of page objects that can be created. This number must be at least the number of Target Pages shown on Document tab.

Max Fonts: the total number of fonts that are contained in the document. Note that a font in PDF includes style variations such as bold and italic, so a typical font may require up to four fonts within the PDF file (Plain, Bold, Italic and Bold-Italic faces).

Max Images: the total number of images stored as JPEGs in a file. Each image is stored as a separate object. Although PDF supports reusing images (i.e. an image can appear on several pages in the document but is represented by a single instance in the file), PDF ePrinter has no way of knowing which images are really the same, so every image is included as a separate object. This is the attribute you will most likely have to adjust, particularly if you create a document with a logo on every page as well as other graphics.

Max Annotations: you should not have to change this value since creation of annotations is not supported in this version of PDF ePrinter.

Max Objects: the total number of PDF objects that can be contained in the PDF file. Each page, font, image and outline is an object, and there are numerous overhead objects generated automatically. This is why this number is so much larger. If you make a major change to any of the other numbers, you should increase this one, as well.

The next four parameters have to do with reflowing horizontal text. As will be discussed further below, the fonts used in a PDF file may not be the same as those used in the original file. Even fonts with the same names (Times and Times New Roman, for example), may have different metrics, or character sizes, between the original and the PDF version. Depending on how the creating application generates the text, this can cause gaps and overlaps in the text when there is a style or font change in the middle of a line. PDF ePrinter provides a solution for this by looking for these gaps and eliminating them. The following parameters control this behavior:

Hor. Text Flow: This parameter turns the option on or off; set it to 1 to turn it on, 0 to turn it off.

HTF Max: This determines the window, in points, in which the function operates. By default, it is set to 72, which means that any text that would overlap or gap within +/- 72 points will be repositioned following the text prior to it. This window is important because you would not want to reposition text on the same line but in a different column. Note: this option will not work with all applications; some print text in different fonts and styles non-sequentially. PDF ePrinter will not be able to guess where the text should go. If overlapping gets worse when this option is on, this is probably the case.

HTF Vertical Tol.: The vertical range, in points, in which text will be considered in the same line as preceding text for horizontal flowing. By default, this value is zero so that text must be on the same baseline in order to be horizontally repositioned. If the value is negative, the vertical tolerance will be set to the current text size plus the positive of the value minus 1. This setting can be used to reflow super- or subscripted text, or text of differing sizes and baselines.

HTF Style Flags: Four control flags for determining which text is considered part of the same sequence for horizontal reflowing. The flags are represented by single letters, of which some, none or all may be present. You do not need to include separators or spaces in the string of flags. The flags are: C (color), S (size), F (font) and (Y) style. When present, the flag indicates that a change in the associated parameter of the text will be considered a change in the text stream, and subsequent text will not be adjusted relative to previous text. This is useful for preventing text of

differing styles from being moved (for example, text in one font will not be repositioned next to a preceding bullet in another font if the ‘F’ flag is present). The ‘Y’ flag considers bold, italics, underline and outline to be a style change.

Horizontal text flow can also be turned on and off within the file using the and the various horizontal text flow parameters can be set using "h" meta-command; see below.

Printer Image Scaling Factor: the StyleWriter driver will generally cause the internal resolution of bitmapped images to be increased to 360dpi, or five-times the screen resolution (72dpi). This value corrects for this. Note that some applications increase the resolution of the entire document, so this factor must be set to 1 for them.

TxRatio Tolerance: this obscure setting allows for the proper handling of scaled text with some applications. This should normally be set to 0, but can be set to 1 for applications requiring it.

Compression Settings

If the ‘Compress PDF file (zlib)’ box is checked, then the page objects within the PDF file will be compressed using the zlib algorithm (embedded TrueType® fonts will also be compressed when this is selected). This algorithm is similar to the zip format. Normally this setting is on and will cause the generated PDF files to be significantly smaller. However, you may want to turn it off if the PDF file will be viewed using software which does not support decompression; Acrobat® Reader™ does provide this support, but some third party software may not.

Faster Picture Rendering

When this option is set, a faster method for converting pictures to PDF is used. Normally, you should leave this option set because it will save time and memory in the conversion process. However, if there are pictures in your document that are rotated or transformed in other ways, you may have to uncheck this option for them to display correctly. In particular, black & white (1-bit) images tend to become corrupted by this process. This option is not compatible with the StyleWriter driver; if you use this driver, you will be warned if this option is left on. Using this option with the StyleWriter may cause images to appear incorrectly or not at all.

Unband Images

This option causes bitmapped images that have been split into individual bands by the LaserWriter driver to be recombined into a single image. For some types of images, this may result in increased compression, particularly if the total image size exceeds the threshold for storing the image as a JPEG (see JPEG Conversion below). This option is generally not necessary with the StyleWriter; images are typically not banded with this driver.

JPEG Conversion

PDF ePrinter stores bitmapped images in the PDF file in either an inline, raw format, or as JPEGs. Generally, handling of large images is more efficient if they are in JPEG format. You can select the cutoff point in image size above which PDF ePrinter will convert to JPEG; by default it is set to 32KBytes. If you set it to zero, all images will be converted to JPEGs. Be careful about setting this value too low; you may need to increase the number of images allowed in the

file. If the value is set high, you may need to allocate more memory to PDF ePrinter to handle the images. This setting can also be controlled within the document by the "j" meta-command. Images that have been inlined will be compressed with zlib, if it is active.

Note: some programs "band" bitmaps, that is, they break them into multiple, horizontal stripes when printing (and some programs, notably Acrobat Reader, print the whole document as a series of bitmaps; these don't make for very good PDF files). This will create a large number of small images out of a large image. In this case, you will definitely want to set this value larger than the size of each band. How do you know if an application has banded its images? When PDF ePrinter writes a JPEG to the file, it will display a message in the progress bar window; if you see a large number of these flashing by on one page, and there are few images on the page, the images have been banded. This message will also indicate the total number of JPEGs created for the document; if this number exceeds the *maxImages* value, you will get an error. You can use the Unband Images option discussed above to handle this problem (PDF ePrinter also displays a message when storing an inline graphic to let you know what is happening). This is normal (unless you crash, in which case you may need more memory to handle the images).

If you are using QuickTime 4 or newer, the JPEG Settings button will be enabled. This will display QuickTime's JPEG Settings dialog where you can set the quality for compressed images. To preserve image quality, set the Quality slider higher. This will result in less compression, but less loss of image clarity. To gain more compression, set the slider to lower quality. Note: if you do not have QuickTime 4 or better, images will still be converted to JPEGs, but you will not be able to control the image quality or size.

Font Mappings

PDF ePrinter 1.5 provides support for referencing and embedding TrueType® and PostScript® Type1 fonts. You can also substitute one font for another, or include a font as a simple bitmap. If you wish to use the font mappings that you have saved, check the 'Use Saved PDF font mappings' box. If you do not have this checked and a font other than the standard or defined Type I fonts appears in the file, you will be prompted to select a substitution font (regardless of whether is checked, if an unmapped font is encountered, you will be asked to select a mapping). Click the Manage Fonts... button to access the PDF Fonts Manager dialog; see the section on |p33=Font Management| below.

Bitmapped Fonts

One of the options available in the PDF Fonts Manager is to embed a font as a bitmapped graphic. If this is selected as a substitution for a particular font, the text of that font will be rendered as a bitmapped graphic by PDF ePrinter and stored inline with the text. This functionality is primarily available to allow unique fonts (such as symbolic fonts) to be displayed accurately in the final document. The drawbacks to displaying text this way are that it makes the file larger (because bitmaps are being stored instead of text), and the text rendered in the font cannot be search, indexed or selected as regular text. Therefore, this option should only be used in limited situations. However, these limitations can be partially overcome by enabling the next option, *Insert hidden text behind bitmaps*. When selected, this will include the original text as invisible characters in approximately the same position as the bitmapped text. This will allow the user to

select and index the text. The key to making this trick work well is to select a hidden text substitution font that has nearly the same metrics as the font being embedded as graphics. See the section on Using the PDF Fonts Manager for more information on how to do this.

When PDF ePrinter renders the text, it allows you to select the resolution of the output bitmap. By default, it is set to 72dpi, which is normal screen resolution. You can select multiples of this resolution up to 5 times normal, or 360dpi. Increasing the resolution will take up proportionally more space, but will cause the displayed fonts to be smoother on high resolution output devices like laser and inkjet printers. If your document will primarily be read on-line, leave the resolution setting low.

Notes: Text stored as bitmapped graphics is inserted inline with the text. This means that it is not stored as a separate object in the output file, so you do not need to adjust the memory value for max images. Also, because it is stored inline, it is compressed using the zlib compression algorithm, if that has been enabled. Inlining of bitmapped text is not affected by the JPEG image size option; these bitmaps are never converted to JPEGs. When bitmapped text is inlined with regular text, there may be small alignment errors and overlaps. This is due to the fact that the bitmapped text metrics are generally different than the normal text's metrics, particularly if the normal text is a substitution font.

Picture Comments

The 'Process Picture Comments' checkbox enables or disables handling of comments embedded within the QuickDraw file. In order to provide access to PostScript Language features not available in QuickDraw, such as rotated text, many applications will embed special commands in their printed output that can be interpreted by the LaserWriter driver to generate PostScript Language. When picture comment processing is enabled, some of the commands will work with PDF ePrinter. In particular, commands for rotating text and graphics, setting fractional line widths, and dashed lines are converted to the equivalent PDF commands. PDF ePrinter will not interpret embedded PostScript language or process polygons as smoothed curves.

You should generally leave this option enabled, unless you find a specific problem with the generated PDF that might be related to picture comment processing. Applications generate and handle picture comments slightly differently, so results may vary. Applications should provide a QuickDraw representation of the PostScript language object (such as a bitmap of rotated text); however, some applications detect the LaserWriter driver an only generate the PostScript Language or picture comment-based version of the output, and not the QuickDraw. In this case, disabling picture comments may cause the object not to appear at all in the output.

Meta-Commands

Meta-commands allow you to embed processing instructions within your document. These are recognized within text fields by their delimiter brackets. Normally, these brackets are '!@' on the front of the command and '@!' at the end of the command. However, you may find a circumstance where this particular combination of character occurs within the text of a file. If this is the case, and you want to use meta-commands in the file, you can change the delimiter.

The right delimiter is always the reverse of the left delimiter; as you type the left delimiter, the right is automatically generated. Note: this means you cannot use standard matching parens such as (), {} or [].

One of the available meta-commands allows you to create intra-document links. Since PDF documents are page numbered from one, you may want to align the page numbers of link with the actual page numbering of your document. If you have an unnumbered cover page followed by the first numbered page, you can set an offset of ‘1’ in this dialog. Then when you use an intra-document link to page 1, it will actually go to the second page of the file (which will be the first page after the cover).

The ‘Enable Meta-Commands’ checkbox must be checked for meta-commands to be processed, If it is not, meta-commands will be ignored. Note: if you are not using meta-commands, leave this unchecked. Processing will be faster and there will be no possibility of some text that has the meta-command delimiter in it accidently being interpreted as a meta-command.

See the Meta-Commands section below for more about using this capability.

Automatic Printer and Application Settings

Some of the parameters above can be set automatically when a job is opened based on both the printer driver and application that generated the job. These are contained in two files located in the Components sub-folder of the PDF ePrinter folder: "Application-specific Settings" and "Printer-specific Settings". You can add entries to these files as necessary to support your applications.

The parameters that can be set are the same for both printers and applications; their names correspond to the AppleScript parameter names given below. The printer parameters are applied first and then the application parameters; this means the application parameters can override the printer parameters (both printer and application parameters are applied *after* the default or last template is applied, if these options are enabled).

The parameters that can be set this way are:

`imagescalefactor`	integer
`txratiotolerance`	0 or 1
`fasterpicts`	0 or 1
`unbandimages`	0 or 1
`processpiccomments`	0 or 1
`htextflow`	0 or 1
`htextflowmax`	integer
`htextverttol`	integer
`htextflags`	string containing C,S,F and/or Y

The files are simple text files with one setting per line. Each setting contains three parts separated by colons(:). The first field is the application or printer name exactly as it appears in the

operating system. In order to support applications which contain a version number in their names, or a family of printers, you can use an asterisk (*) in the right-most position of this field as a wildcard: "Application*" will match "Application 1.0" and "Application 2.0". The second field is the parameter name from the list above. The last field is the value to which the parameter should be set. You can also include a comment line by making the first character of the line a percent sign (%).

Managing Files

This section describes the commands available under the File menu for saving print jobs and working with templates. Some of these commands can also be accessed from buttons within the job window and from AppleScript.

Saving and Opening Saved Jobs

PDF ePrinter automatically opens a job window for each print job sent to the desktop printer folder it is monitoring. If you don't want to convert this file to a PDF immediately, you can save it as an ePrinter spool file and open it later. Since PDF ePrinter deletes the file from the desktop printer queue after it converts it to PDF, this is the only way to reuse a spool file (when you convert from a saved spool file, PDF ePrinter *does not* delete the file).

To save a job as an ePrinter spool file, click the Save and Defer... button in the job window, or select Save and Defer... from the File menu. This will display the standard file save dialog for you to name the file. When you save the file, the job is removed from the queue and the job window is closed.

You can open a saved file by selecting Open... from the File menu and selecting the file in the standard file open dialog. You can also choose a file from the Recent Files menu. The four most-recently accessed files are listed here. Note that when you save a file, its name is added to the top of the list so that you can immediately re-open it if you want to process it. Also, when you open a file this way, the Save and Defer... options in the job window and File menu become Save As... options, allowing you to resave the file under a different name.

Cancelling Jobs

You can choose to cancel a job by selecting Close from the File menu, clicking the job window's close box, or clicking the Cancel button. Doing any of these will cause you to be prompted to remind you that this action will permanently delete the job from the queue.

Templates

A PDF ePrinter template file stores all of the settings on the various tabs of the job window. Saving these settings as a template will allow you to quickly restore the configuration for a specific job or for any job that contains similar formatting.

You can save a template at anytime when a job window is open by selecting Save Template As... from the File menu and providing a name and location. If you previously loaded a template, you can resave it to the same file by selecting Save Template.

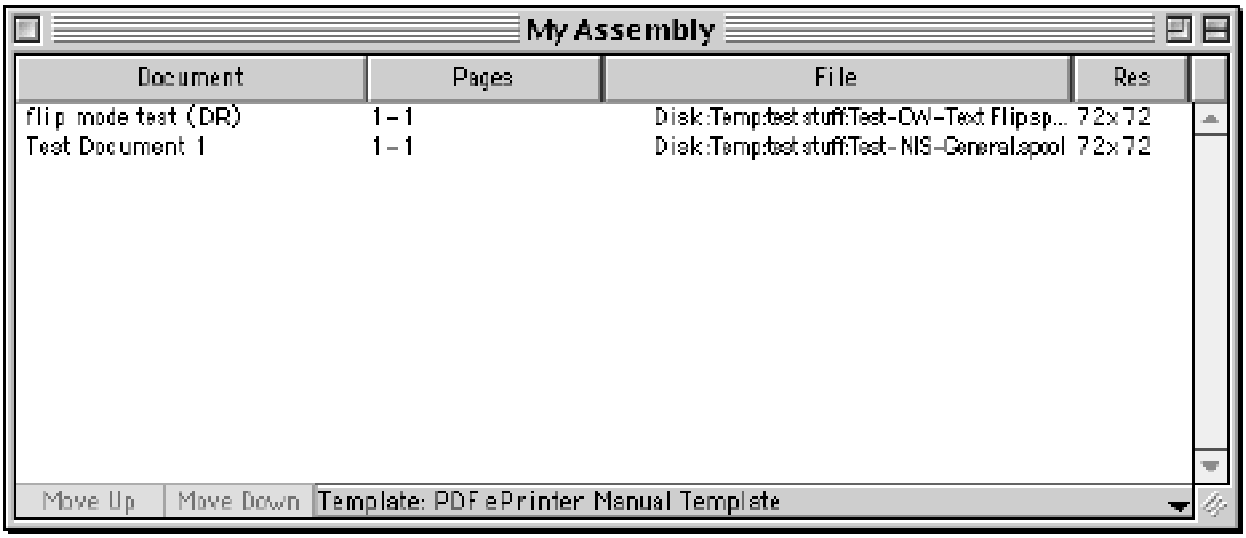
Once you have saved a template, you can open it by choosing Load Template... from the File menu and selecting the template file. The Recent Templates menu maintains a list of the four most-recently used templates for quick access.

You can also have a template be automatically opened for every job by specifying it in the Preferences dialog (reached by choosing Preferences... from the Edit menu). This procedure is described above in the Configuring PDF ePrinter section.

Note that a saved template stores all of the parameters on the tabs of the job window including the JPEG settings for image compression. Even if a parameter is blank, it is saved so that a blank parameter will overwrite (clear) anything that may be entered into a field prior to the template being loaded. If you want to work from a template and then make modifications to the settings, first load the template and then make the changes. You can also load a template from AppleScript and then make individual parameter changes programmatically.

Assemblies

PDF ePrinter allows you to combine several saved jobs into a single PDF file by creating an assembly. To create a new assembly, select New... from the Assemble sub-menu of the File menu. This will open a blank assembly window. You can add saved job files by selected Add... from the File menu; you can also drag saved jobs from the Finder to an open assembly window.



Assembly Window

The assembly window shows the document's title, the range of pages to be included in the assembled PDF file, the path to the source file, and the resolution at which the job was captured. A job can be included more than once. You can change the order of jobs by selecting one or more jobs and clicking the Move Up and Move Down buttons at the bottom of the window. You can also remove entries in the window by selecting Delete from the Edit menu, and duplicate one or more entries by selecting Duplicate in the Edit menu.

By clicking in the Pages column for an entry, you can edit the included pages specification; it uses the same format as the Use Pages entry on the Document Tab of the Job Window. If you enter an illegal page specification, the text for the entry will be bolded. Double-clicking a row will allow you to select a new source file for the entry.

You can choose a Template file to associate with the assembly from the drop-down list at the bottom of the window. Note that only one template file can be associated with the complete assembly; if one of the files in the assembly requires some special processing settings, you will have to set these with meta-commands embedded within that job, if possible.

You can save an assembly to an assembly file by using the Save Assembly As... command in the File menu. When you open a saved assembly file, PDF ePrinter will check to see if the job files referenced in the are still available and the page ranges are still valid. Missing job files will be indicated by italics in the row for the missing file. Bold text in the pages column indicates an invalid pages entry. These errors must be corrected before an assembly can be converted into a PDF job.

The Res column indicates the resolution at which the job was captured. You should try not to mix resolutions in a single file; results will vary.

When you have completely specified the assembly, choose Create Job... from the File menu. If there are no errors in the assembly specification, the files will be combined into a single job file which will be opened in a new Job Window. If a template was selected, it will be applied automatically. From this point, processing into a completed PDF file is as usual.

Assembly File Format

Assembly files are simple text files; this allows you to generate an assembly specification in another application and automate the task of generating combined PDF files. Note: only the opening of an assembly file and creation of a job can be automated from AppleScript; if you want to produce assemblies on the fly, you must generate the assembly file separately and open it in PDF ePrinter from AppleScript.

Each line is plain text terminated by a carriage return character (ASCII 13). For all but the first line, entries are tab-delimited (ASCII 9). The first line is the full path to the template file (blank line if none assigned). Each line after that is one entry from the jobs list in the order they appear in the assembly window list. The first field is the full path to the job file. This is followed by a tab character and the text of the page ranges specification. (The other fields that appear in the assembly dialog are read from the job file).

Note: to open an assembly file saved from an application other than PDF ePrinter, hold down the Option key while selecting Assemble, Open... from the File menu.

Advanced Features

Using Meta-Commands

PDF ePrinter allows you to embed instructions within your document that are interpreted by the processor as the file is converted to PDF. This provides a fine control over formatting of the document, as well as a way to embed conversion instructions within the actual document so that, as the document is changed, it will require less reconfiguration to convert to PDF.

All meta-commands are contained within the document as text. Commands are placed between two delimiters, or text sequences, that identify the command to the processor. By default these are !@ to start a command and @! to end a command. However, in the event one of these sequences appears within the text normally, you will have provide a different delimiter sequence. This is done on the Advanced tab of the job dialog as described above. The closing sequence is always the reverse of the opening sequence. Note that you must also check the box to enable meta-command processing on this tab; otherwise, commands will be ignored and will appear in the output file unprocessed.

A command can be embedded within any standard text (as long as the text is represented to QuickDraw as text and not a graphic). For most commands, as they are processed, their text is removed from the output. Because of this, the spacing of text after the command may be affected. Some of the commands can appear anywhere on the page because they affect the entire page. If possible, these can be placed in text elements outside the normal flow of text, or in a line at the end of the page. Other commands must appear inline because they are associated with the text itself. In these instances, it may be necessary to create a trial output file and make adjustments to word placement or line lengths to compensate for the text incorporated into the command, if layout is critical. You can also try using the horizontal text flow option which may help reflow the text to avoid gaps (the previous line is a good example of this problem! It has been left like this as a demonstration. A possible fix would be to push the right margin out for just that line to cause the next word or two to be on the line to fill the gap). The effectiveness of any solution will depend on how the text is generated by your application!

Be careful about changing text styles within a command; this change will not be caught by the processor. Also, do not break commands across lines or text elements. If a command is not working properly in the output, it is probably because it has been broken into multiple text elements when the document was rendered with QuickDraw during printing.

The Meta-Commands

The following is a list of meta-commands. Each is shown with the standard delimiters. Within each command, text in italics represents variables that must be replaced with values when you use the command. These variables are described in the command summary following each command. Variables that are numeric are represented with the number sign (#); do not include the number sign in the actual command. Commands are case-sensitive; all of the commands use a lowercase letter after the left delimiter. Also, you should not include whitespace except within text variables.

Transition

`!@t#1,#2,#3,#4,#5@!`

Sets a page transition. The first variable is the transition type and is one of the following: 0=no transition, 1=split, 2=blinds, 3=box, 4=wipe, 5=dissolve, and 6=glitter. The second value is the duration of the effect in seconds. The third variable is the direction of the effect in degrees where 0 is to the right, 90 is up, etc. The fourth setting specifies the orientation of the transition borders (0=vertical, 1=horizontal) for the split and blinds transitions. The last variable determines whether the effect for the box and split transitions is out (0) or in (1).

Duration

`!@d#1@!`

Sets the duration of the page in seconds. If a page has a duration setting, it will automatically advance when the time expires. This is useful for creating automatic slide shows (you can use the Fullscreen page mode parameter on the Viewer tab of the job dialog in conjunction with this command to create an automatic slide show).

Outline Entry

`!@o#1,#2,mode=text@1`

Creates an outline entry. The first variable is the level. The highest (root) level of the outline tree is numbered one (1). Outline entries should be added such that a proper ordering of increasing levels is maintained; the command processor will enforce this out of necessity by demoting an entry to one greater than the previous entry if a level is skipped. The second variable determines whether the outline subtree for this entry is open (1) or closed (0) when the outline is first displayed; this is equivalent to the Open parameter in the auto-outline settings. The final variable is text containing the mode. The modes are the same as provided in the auto-outlining settings: `Null`, `fit`, `fitH`, `fitV`, `Y`. The text after the equals sign is the heading that will appear in the outline entry. This text is also inserted back into the output text in the place of the command.

This command will work in conjunction with auto-outlining. Levels defined here are equivalent to the first five outline styles defined on the Auto-outline tab. The rule for demoting an entry applies; if an entry created by either auto-outlining or the meta-command would cause a level to be skipped between the last entry made (by either method), the new entry will be demoted to one level less than the last entry.

Intra-document Hyperlink

!@p#*1*=*text*@!

Creates a link to another page within the document. The first variable is the page number to which the jump should be made, and the second variable (after the equals sign) is the hypertext (the text that will be considered "hot.") This text will be passed back in place of the command in the document. The page number is absolute to the first page of the output document, numbered from one. If you want to make this number relative to the first numbered page of the document, enter the different between the first page (the cover) of the document and the first numbered page in the ‘Link to Page’ offset field of the Advanced tab of the job window. This offset will be added to the page number provided as the first variable to this command to get the actual page number within the PDF document.

Extra-document Hyperlink

!@u*text*@!

Creates a hyper link to a URL. The text variable is the complete URL; this text will be passed back to replace the command in the output document. Remember that the command cannot be split across lines, so it may not be possible to include long URLs in a document.

Horizontal Text Flow

!@h#*1*@!

Turns horizontal text flow control on or off. Set to 1 to turn on, 0 to turn off.

!@hm#*1*@!

Sets the horizontal text flow maximum value.

!@hv#*1*@!

Sets the horizontal text flow vertical tolerance value.

!@hf*text*@!

Sets the horizontal text flow style flags.

JPEG Conversion Size

!@j#*1*@!

Determines the maximum size an image can be before it is converted to a JPEG instead of being inlined. The value of #1 is in KBytes, so 32 is 32K. If this is set to zero, all images will be converted to JPEGs. Be careful using this with "banded" images; this may result in a large number of image objects being created. This only applies to all subsequent images. Note that some applications may not generate images and text in the same order as seen on the page. See the section on JPEG Conversion for more information.

JPEG Quality

!@q#*1*@!

Sets the quality for subsequent JPEG images. #1 is between 0 (maximum compression, minimum quality), and 1024 (minimum compression, maximum quality).

AppleScript Support

PDF ePrinter can be controlled by AppleScript which allows it to be used in automated workflows which produce PDF. Additionally, because PDF production parameters can be set in code, on-the-fly formatting is possible for applications which need to respond in real time to user requests, such as web-based systems.

The process for using PDF ePrinter from an AppleScript is similar to the manual process. First, a file must be printed with a production application to the ePrinter desktop printer; alternately, you can use an existing file previously saved as a spool file using the Save and Defer command. If you are using LaserWriter 8.7 (available with Mac OS 9), and an application that supports scriptable printing, you can control many details of the print job. Otherwise, you will have to simply print the document using the application’s default print command via the PrintDocument Apple event. Once the document has been printed and is in the queue, start the PDF ePrinter application. Assuming there were no other documents in the queue, the most-recently printed document will open in a job window. At this point you can use AppleScript commands to save the file as a spool file, open and apply a template, change individual parameters, and finally create the PDF output file. When you are done with PDF ePrinter, send it the Quit command.

With AppleScript, you can also handle multiple jobs simultaneously. Each job is assigned a job ID which you then use in subsequent commands to uniquely address it.

The following is a list of commands with their AppleScript syntax and a description. Items in italics are variables that should be replaced with actual values in a script. All commands should be contained within a tell block for PDF ePrinter:

```
tell application "PDF ePrinter"
     commands...
end tell
```

Note: The `GetParameter` and `SetParameter` commands are no longer supported as of version 1.4.

The PDF ePrinter Object Model

The object model defines the organization of entities within the PDF ePrinter application such that they can be addressed from an AppleScript. There are three entities arranged hierarchically in the object model: the application, the job and the outlinestyle.

The application object is the root of the hierarchy. It contains several application-wide properties, as well as a list of active jobs. The properties of the application are those found in the Preferences dialog.

Each job represents an individual print job or spool file ready for conversion to PDF. The properties of a job are those found on the tabs of the job dialog. A job is referenced by its job ID number, which is uniquely assigned when the job is created. You can get the job ID when you open a job, or by getting the ID of the currentjob, which is the front job in the list.

Each job also has a list of up to five outlinestyles, each of which has the properties of one row on the Auto-outlining tab of the job dialog. Each outline style is referenced by its numeric index in the range of 1 to 5.

Application Object

Properties

`watchfolder`	string
`bringtofront`	boolean
`xprocessor`	boolean
`defaulttemplate`	string
`lowmemorycheck`	boolean
`memorywarning`	integer
`nouserinteraction`	boolean
`currentjob`	integer
`askdestination`	boolean
`joblist`	list of integer, READ ONLY
`destinationfolder`	string
`fontsubmode`	string (must be one of Manual, Automatic, Suggest)
`embedTTF`	boolean
`defaultsubfont`	string (in the form fontname;encoding)

`open file "`*`file-path`*`"`
Opens the named file. The file may be either a previously saved spool file or a template file. Note that you do not need this command if you have issued a print command from another application which causes a document to be printed to the ePrinter desktop printer folder because the file will be opened automatically once PDF ePrinter is started. You can use this command to open and apply a previously saved template file to the current job. Note: this command is equivalent to the OpenDocument AppleEvent and does not return a job ID. Use the currentjob property after calling this to get the job ID. Use OpenJob to return a job ID.

`DocumentReady`
Returns `true` if a job window is open and ready to process, `false` if not. Use this to determine if a job printed from an application has been passed through the queue and is ready to process. The other commands (except for open and quit) will return errors if no job window is open.

`GetNextJob`
Checks the printer queue and loads the next file, if present. Returns the job ID for a new job as an integer, 0 if the queue was empty.

`JobExists` *`jobID`*
Returns true if the referenced job exists (is active, false if not. This can be useful for determining if a job has finished processing; if a known job no longer exists, it has completed processing and the PDF file is ready.

`OpenAssembly file "`*`file-path`*`"`
Opens an assembly file in a new assembly window. Returns an assembly ID if successful, or 0 if the operation failed.

`AssemblyExists` *`assemblyID`*
Returns TRUE if the assembly identified by the *assemblyID* is open, FALSE if it is not.

`Close` *`jobID|assemblyID`*
Close the job or assembly window identified by the given ID.

Job Objects

Properties

Document Tab

`sourcepages`	Integer, READ ONLY
`targetpages`	Integer, READ ONLY
`usepages`	String
`creator`	String, READ ONLY
`title`	String
`subject`	String
`author`	String
`keywords`	String

Viewer Tab

`pagemode`	normal/outlines/thumbs/fullscreen
`nonfullscreenpagemode`	normal/outlines/thumbs
`pagelayout`	single/1column/2columnleft/2columnright
`hidetoolbar`	Boolean
`hidemenubar`	Boolean
`hidegui`	Boolean
`fitpagetowindow`	Boolean
`centerwindow`	Boolean
`generatethumbs`	Boolean

Auto-Outline Tab

`enableautooutlining`	Boolean
`outlinestylescount`	Integer, 0 to 5

Advanced Tab

`maxpages`	Integer
`maxfonts`	Integer
`maximages`	Integer
`maxannotations`	Integer
`maxobjects`	Integer
`usecompression`	Boolean
`usefontmappings`	Boolean
`bitmapfontresolution`	Integer, one of 72,144,216,288 or 360
`converttojpegsize`	Integer, one of 0,2,4,8,16,32,64 or 128
`processpiccomments`	Boolean
`enablemetacommands`	Boolean
`metaleftbracket`	String
`linktopageoffset`	Integer
`jpegsize`	Integer

`jpegquality`	Integer, 0 to 1024
`htextflow`	Boolean
`htextflowmax`	Integer
`htextverttol`	Integer
`htextflags`	String
`fasterpicts`	Boolean
`unbandimages`	Boolean
`imagescalefactor`	Integer
`hiddentext`	Boolean
`txratiotolerance`	Integer (0 or 1)
`jobid`	Integer

Commands

`OpenJob file "`*`file-path`*`"`
Opens the saved spool file at the given location and returns a job ID as an integer. Only opens spool files; use ApplyTemplate to load a template file.

`ApplyTemplate file "`*`file-path`*`"`
Applies the template file at the given path to the job being addressed.

`save file "`*`file-path`*`"`
Saves the active document to the named file. This has the same effect as the Save and Defer... command, so the job will be removed from the queue and the job window closed. You can use the Open command to immediately reopen the file for processing, if necessary.

`CreatePDF file "`*`file-path`*`"`
Converts the current file to the specified PDF file and closes the job window.

Notes:
1. You must set the value of ‘outlinestylescount’ to match the number of styles you want. You can do this before or after setting the individual table entries.
2. "targetpages" is automatically recalculated when "usepages" is changed
2. Commands that take a file-path as a string can also take an alias

Outlinestyle Object

Properties

`font`	String (must be a valid font name)
`size`	Integer
`bold`	Boolean
`italic`	Boolean
`open`	Boolean
`mode`	null/fit/fitH/fitV/Y

Assembly Object

Commands

`CreateJob`

Converts the assembly into a job which then opens a new Job Window.

Sample Script

The following script prints from an application and then processes the job into a PDF file. This script assumes the currently active printer is the ePrinter desktop printer; if it is not, you would have to set that first (using the Desktop Printer Manager scripting addition).

```
tell application "Scriptable Text Editor"
	open file "Hard Disk:Temp:My Text File"
	print window 1
	quit saving no
end tell

tell application "PDF ePrinter"
	activate
	repeat until DocumentReady
	-- just waiting for the queue -- might want to put a timeout here
	-- you could also use GetNextJob, but be careful
	-- about assigning the JobID returned by it without
	-- first checking to see if it was valid (not zero):
	--   set myJobID to GetNextJob
	--   if myJobID = 0 then
	--         set myJobID to currentjob
	--   end if
	end repeat
	set myJobID to currentjob

	tell job myJobID
		set author to "My name goes here"
		set numberOfPages to targetpages
		set the font of outlinestyle 1 to "Helvetica"
		set outlinestylecount to 1
		set pagemode to fullscreen
		set maxpages to 200
		set enablemetacommands to true
		set metaleftbracket to "!@%"
		set jpegquality to 1024
		CreatePDF file "Hard Disk:Temp:myPDF.pdf"
	end tell
	repeat until not (JobExists myJobID)
		-- need to wait for the job to complete
	end repeat
	quit
end tell
```

Customizing the ePrinter PPD File

When you install PDF ePrinter, a PostScript Printer Description (PPD) file is added to the Printer Descriptions folder within the Extensions folder of your System folder. This is s text file that describes the capabilities of the ePrinter "printer." The LaserWriter driver uses this file to display the Page Setup and Print dialogs. You can make changes to this file to provide additional or different options in those dialogs. Information on the PPD file format is available in Adobe Technical Note 5003 which can be found on Adobe's web site. Note that not all features in a PPD file are supported by the LaserWriter driver or PDF ePrinter.

To make a change to the PPD file, you must open it in a text editor such as SimpleText, if the file is small enough. Make your changes and save the file back to the Printer Descriptions folder. Then you must follow the instructions for Using the Apple Desktop Printer Utility to create a new desktop printer named ePrinter (delete the original first). Be sure to use the ePrinter PPD you edited in this process. This step is required to allow the LaserWriter driver to produce a Parsed PPD file (which is stored in the Preferences folder) from the new PPD. Note that you must use this installation and not the PDF ePrinter utility because it installs a PPD and parsed PPD for the default ePrinter installation.

The main reason you might want to modify the PPD file is to add custom paper sizes. In fact, other than some header information, the only entries in the default ePrinter PPD are for the basic paper sizes. To add a new paper size, simply follow the format of the existing entries. You will have to give the paper size a name (and use it throughout the file). You will also need to know the size of the paper in points. This will be the size in inches multiplied by 72. The ImageableArea and PaperDimension settings contain a string of four numbers enclosed in quotes. These numbers (all in points) are the left edge (normally 0), the top edge (normally 0), the width and the height.

Note that the PPD file is only associated with the LaserWriter driver; changing it will have no effect on the StyleWriter driver.

Other Printer Drivers

You can try other printer drivers with PDF ePrinter. In order to work, a driver must produce (or, more correctly, provide access to) a standard spool file. This file must be formatted as specified in Apple *Technote 1097: Desktop Printing Revealed.* The printer driver should have some type of pause function to prevent it from processing the spool file before PDF ePrinter.

To use a different driver, make it the default driver and set PDF ePrinter's watch folder preference to point to the folder where the driver stores its spool files. Then try printing and examine the results. If PDF ePrinter doesn't see the spool file, it may not have the correct creator and type (which may mean it is not the correct format). If you get a spool file error, the format is incompatible with PDF ePrinter.

If the PDF file produced has problems with bitmapped graphics, you may need to set the Printer image scaling factor in the advanced tab. For the StyleWriter, this needs to be set to 5 (for most applications) which is the printer's resolution (360dpi) divided by 72dpi.

Font Management

The promise of PDF is that your document will appear to the reader exactly as you intended. Unfortunately, this is often unachievable because of fonts; if the user's machine does not have the fonts you used, the document will probably not look the same. PDF attempts to overcome this by allowing fonts to be embedded and by supporting multiple types of fonts. As of version 1.5, PDF ePrinter provides three ways to handle fonts: embedding or referencing outline fonts, substituting, or embedding as a bitmapped graphic.

Depending on which font substitution mode you have selected in the Preferences dialog, PDF ePrinter will either automatically handle fonts, make suggestions to you about font substitutions, or leave all of the work to you.

Several fonts are supported by default; these are the core fonts and they do not need to be explicitly embedded in your document to appear correctly on another person's computer.

The built-in fonts are:

Helvetica	Helvetica-Bold
Helvetica-Oblique	Helvetica-BoldOblique
Times-Roman	Times-Bold
Times-Italic	Times-BoldItalic
Courier	Courier-Bold
Courier-Oblique	Courier-BoldOblique
Symbol	ZapfDingbats
AvantGarde-Book	AvantGarde-BookOblique
AvantGarde-Demi	AvantGarde-DemiOblique
Bookman-Demi	Bookman-DemiItalic
Bookman-Light	Bookman-LightItalic
Helvetica-Narrow	Helvetica-Narrow-Oblique
Helvetica-Narrow-Bold	Helvetica-Narrow-BoldOblique
NewCenturySchlbk-Roman	NewCenturySchlbk-Italic
NewCenturySchlbk-Bold	NewCenturySchlbk-BoldItalic
Palatino-Roman	Palatino-Italic
Palatino-Bold	Palatino-BoldItalic
Helvetica-Condensed	Helvetica-Condensed-Bold
Helvetica-Condensed-Oblique	Helvetica-Condensed-BoldObl
ZapfChancery-MediumItalic	CPDF-Monospace
CPDF-Smallcap	

Additionally, the following Asian fonts are available (each has a plain, bold, italic and bolditalic face):

MHei-Medium (Traditional Chinese)
MSung-Light (Traditional Chinese)
STSong-Light (Simplified Chinese)

HeiseiKakuGo-W5 (Japanese)
HeiseiMin-W3 (Japanese)
HYGoThic-Medium (Korean)
HYSMyeongJo-Medium (Korean)

You can also include other Type 1 fonts by providing the necessary .PFM (metrics) and .PFB (font program) files. These files are typically used on Microsoft® Windows™ computers and may be supplied with the font you obtained. However, if you do not have these files, but do have the LaserWriter font file containing the POST resources describing the font, PDF ePrinter can convert it to a .PFB file. You will still need to obtain a metrics file; many of these are available from font vendors on their web sites. If you cannot get a .PFM file, PDF ePrinter can convert a .AFM file for a font to .PFM format.

You can also embed TrueType fonts. PDF ePrinter requires that you first prepare a TrueType font and provides an assistant for doing this. When you prepare a TrueType font, PDF ePrinter generates a metrics file, a font stub .PFB file, and a .TTF file containing the font program. If you choose to embed the font, the .TTF file is stored in the PDF document in a post-processing step.

Note: PDF ePrinter does not try to determine the legal status of the fonts you choose to embed. You must ensure all fonts you select from embedding are properly licensed to you and you are legally allowed to embed them. PDF ePrinter does not support font subsetting.

In general, you should consider who will be reading your document and whether you need to embed fonts. For documents that will be used by persons have access to the same fonts as you have, you can simply reference the font and the document should appear without problems. For fonts that are essentially the same, but have different names (such as Courier and Courier New), use substitution to replace the font with a core font instead of embedding it. If you use only a few characters from a font, particularly a symbolic font, consider embedding it as a graphic.

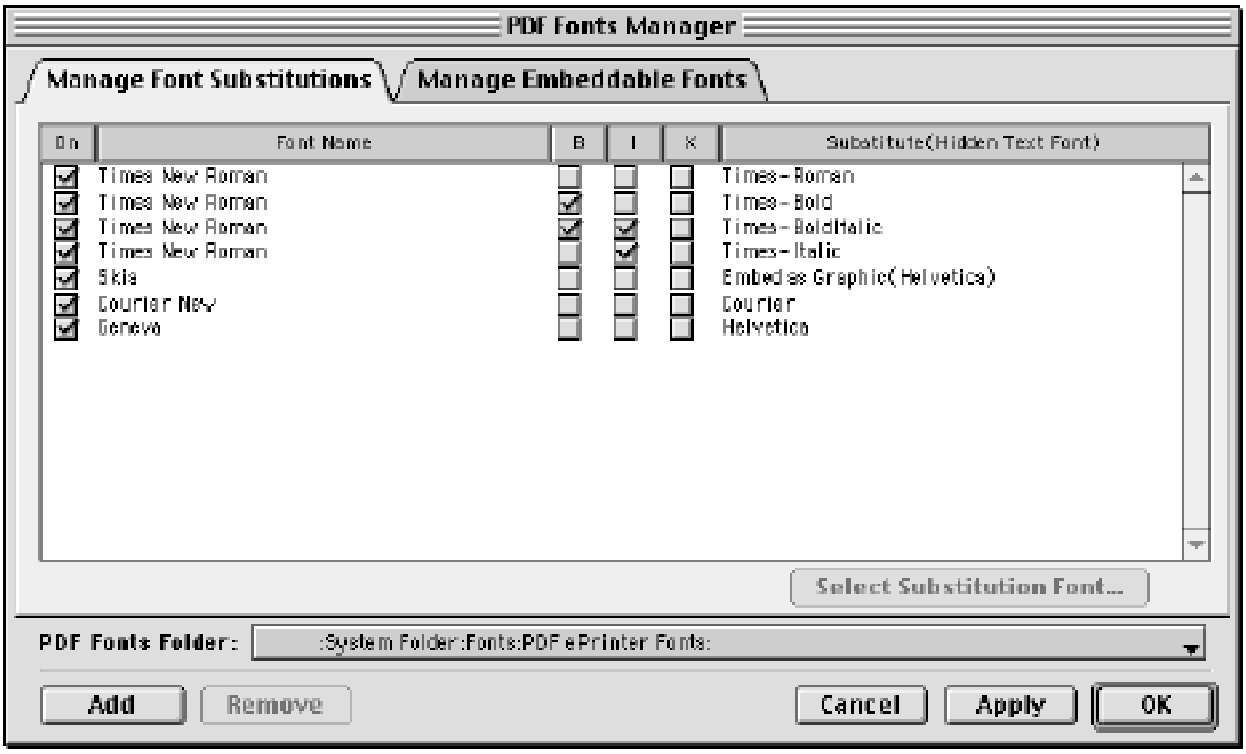
Using the PDF Fonts Manager

The PDF Fonts Manager is accessible from the File menu by selecting Manage Fonts.... This displays the Manager's dialog box which contains two tabs.

With either tab visible, you can select a folder in which to store the font files required by PDF ePrinter. By default, your system Fonts folder is used. Because PDF ePrinter may generate up to three new files for each font you choose to embed, you may want to store all of these files in a sub-folder. Use the popup menu to select a different folder. If you change the folder, be sure to move the .PFB, .PFM and .TTF files created by PDF ePrinter (or supplied by you) that are being used to the new folder.

The first tab, Manage Font Substitutions, allows you to identify which of the default or other defined Type 1 or TrueType fonts you want to be automatically substituted when the PDF file is created (Note: you must have the 'Use saved PDF font mappings' checkbox on the Advanced tab

of the job window checked in order for these substitutions to be made automatically. Otherwise, you will be prompted for each substitution.)



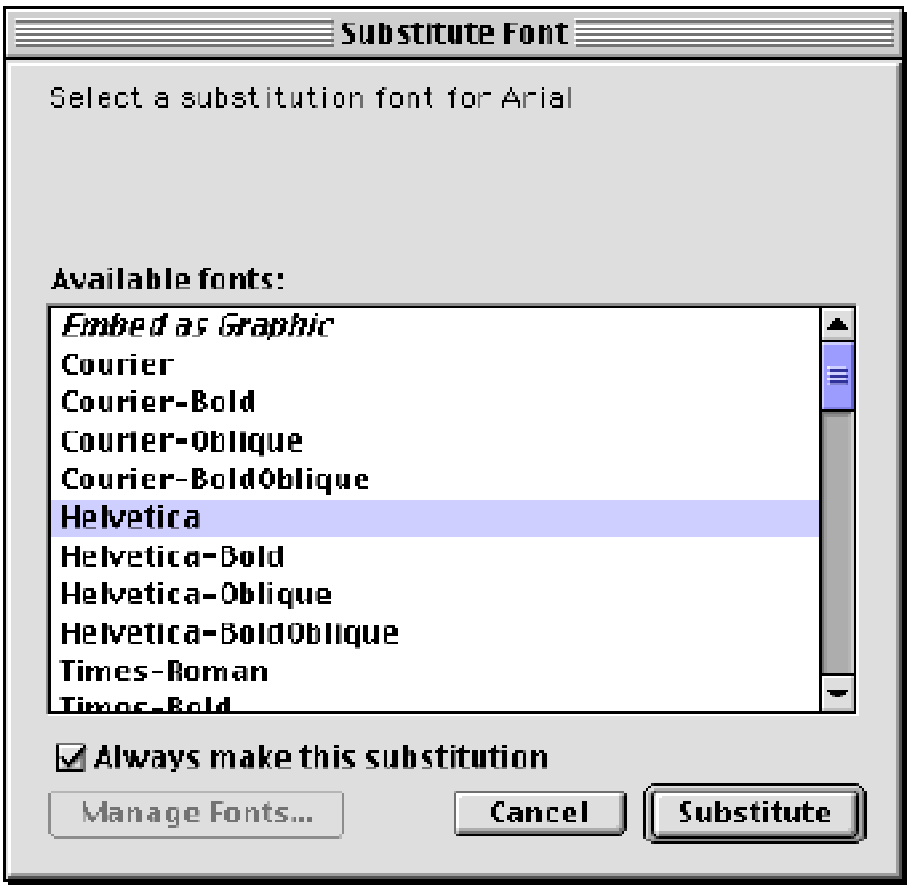
The PDF Fonts Manager Window with the Manage Font Substitutions Tab selected

The columns in the list box are:

On:	When checked, this substitution is active
Font Name:	The font for which the substitution will be made
B:	Indicates the bold style
I:	Indicates the italic style
X:	Indicates all styles
Substitute (Hidden text font):	The font that will be substituted. If this is *embed as graphics*, the name of the font that will be used for hidden text substitution appears in parentheses.

Because the substitution fonts have different font faces for each of the styles Bold, Italic and Bold-Italic, you must indicate which font goes with which style. So, although there may be only a single font name for all of these styles in the application you used to create the file, you would have to create four substitutions: one for the plain font, with neither B or I checked, one with B checked, one with I checked, and one with both B and I checked. The X indicator can be used to identify a substitution that will be made for any style (i.e. if there is only one Type I face). This is most often the case with symbolic fonts.

Click the Add button to add a new entry and type the name of the font in the box. The check the appropriate style(s) and click the row to select it. You will then be able to click the Select Substitution Font... button to choose one of the available fonts. This process can also be automatically accomplished by checking the "Always make this substitution" button when you are prompted during creation of a PDF to select a substitution font (but only if you are using the saved PDF font mappings, otherwise, the selection is not saved beyond that run).



The Font Substitution Selection Dialog

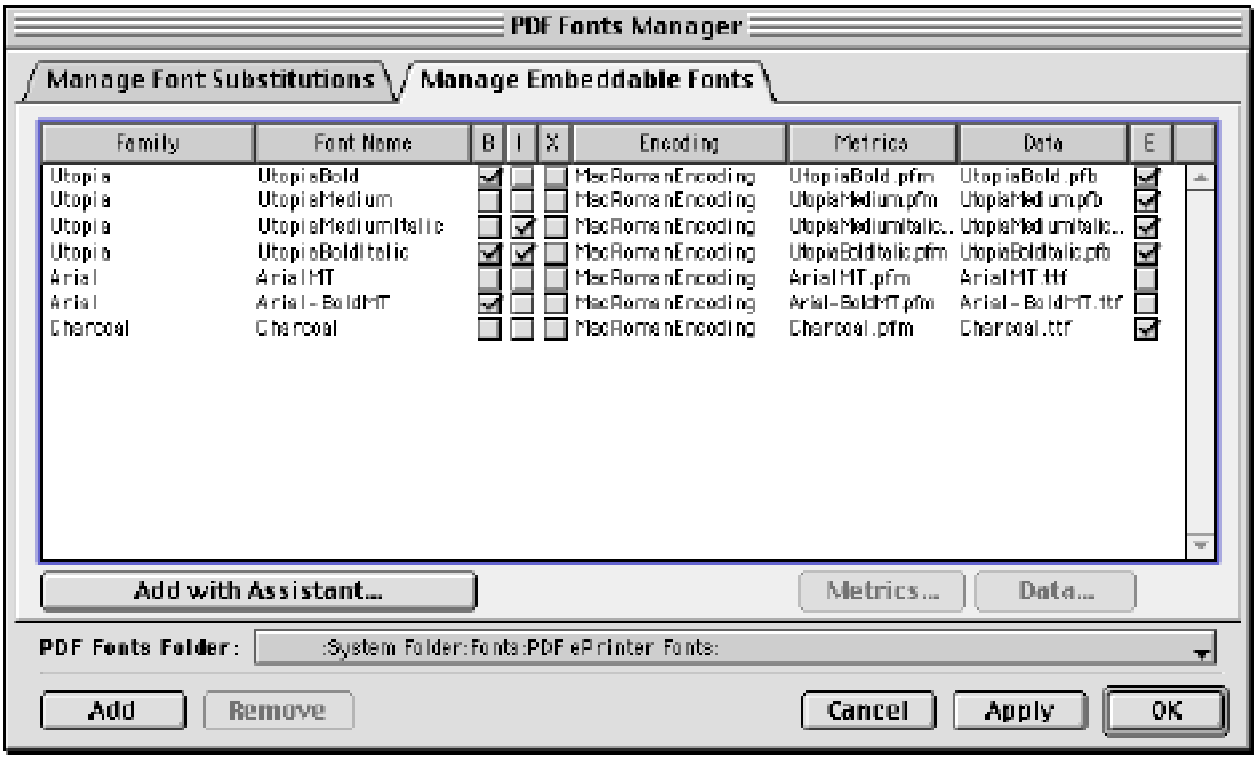
The font substitution selection dialog is displayed when you click the Select Substitution Font... button. It lists all of the fonts that are available to substitute. It also provides a substitution called "Embed as Graphic" which renders the text in that font as a bitmapped graphic and stores it inline; see the Bitmapped Fonts section under the discussion of the Advanced Tab above.
This dialog includes a checkbox to 'Always make this substitution.' If checked, the substitution will be added to the master list. If you have selected to use the saved PDF font mappings (on the Advanced Tab), this substitution will be added to that list and saved for future jobs. If this was not selected, this mapping will only be used during this job. If you do not check this box, you will be prompted every time text in this font is encountered within the document.

If you select *Embed as graphic* for the font, this dialog will be presented again requesting the font to be used for hidden text substitution. When you substitute a font, if you select Embed as

Graphic, the font will be reproduced as a bitmapped graphic in the document. This will preserve the document's original appearance at the expense of some additional size to store the bitmaps. You can also include the original text as hidden characters behind the bitmaps. This will allow the user to select the text, as well as allow the text to be searched and indexed. When you select Embed as Graphic, you will immediately be presented with the substitution dialog again requesting the font to be used for hidden text. This must be on of the available built-in or Type 1 fonts. Select a font with similar metrics (character widths) to the original for the best effect. One way to check this is to compare two lines of the same text in each of the fonts and choose the closest match in size, as demonstrated below:

This is the bitmapped font followed by Helvetica and Times.
This is the bitmapped font followed by Helvetica and Times.
This is the bitmapped font followed by Helvetica and Times.

Helvetica is the closer match, in this case. Of course, if the font is not proportionally spaced, you should use a monospaced font such as Courier. Note in the above example how you can select the bitmapped text with the text selection tool because of the hidden text.



The PDF Fonts Manager Window with the Manage Embeddable Fonts Tab selected

The second tab, Manage Embeddable Fonts, allows you to add Type 1 and TrueType fonts to the list of available substitution fonts. The columns in the listbox are:

Family: the name of the font family as you would see it in the system font list. You can have multiple entries for each family representing the different faces (plain, bold, italic and bold italic).

Font Name: the specific name of the font, normally the PostScript name

B,I,X: when checked, indicate that the font is either bold (B), italic (I) or both, or could represent any face (X). A font cannot be bold and/or italic and have the X box marked.

Encoding: the character set used in the font. Almost all will be MacRomanEncoding. If you suspect the font is another encoding, use only one of the encoding names specified in the ClibPDF manual.

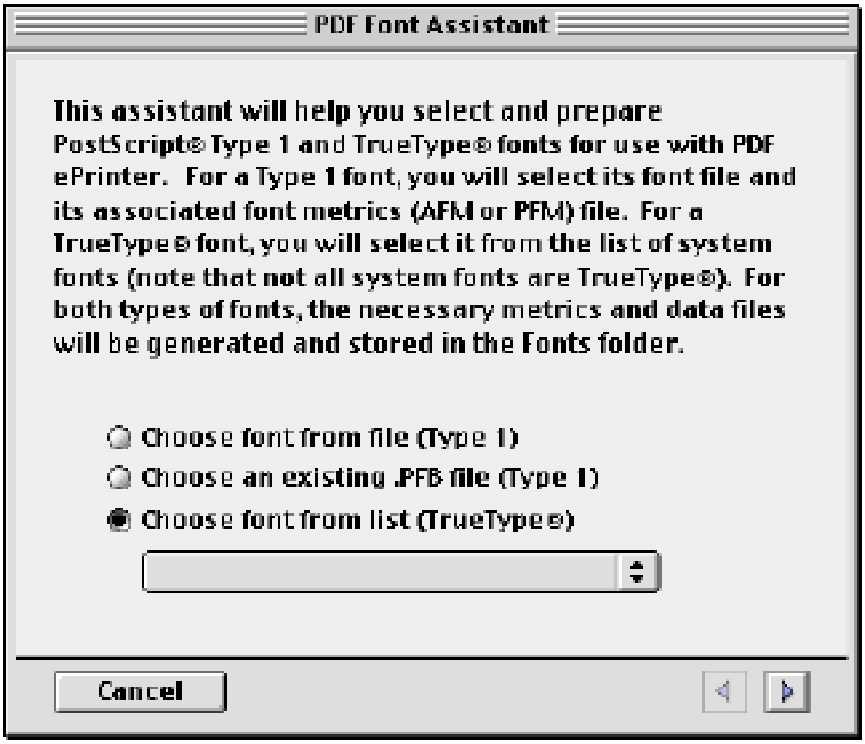
Metrics: the metrics file, always a .PFM file.

Data: the font's data file containing the font program. A .PFB for Type 1 fonts, and a .TTF for TrueType fonts

E: checked if the font is to be embedded. Fonts that aren't embedded will be referenced. If they are available on the user's computer, they will be used. Otherwise, the PDF viewer may attempt to substitute a font using the original font's metrics to maintain as close a rendition of the original as possible. Embedding a font adds about 50KBytes to a PDF file. TrueType fonts are compressed if file compression is on (Type 1 fonts are encrypted and do not compress well, although the size of a TrueType font after compression is generally the same as a Type 1 font).

To change a metrics or data file for an item, select it in the list and click the appropriate button. You can also select multiple items and click the Remove button, or click Add to create a new entry manually.

Although you can enter this information manually, you should use the Fonts Assistant by clicking the Add with Assistant... button. When you do, the following dialog is presented:



The PDF Font Assistant Dialog - First Window

From here, you select the type of font you want to prepare for use with PDF ePrinter and the Assistant will step you through the process, ending by adding the new information to the Fonts Manager window.

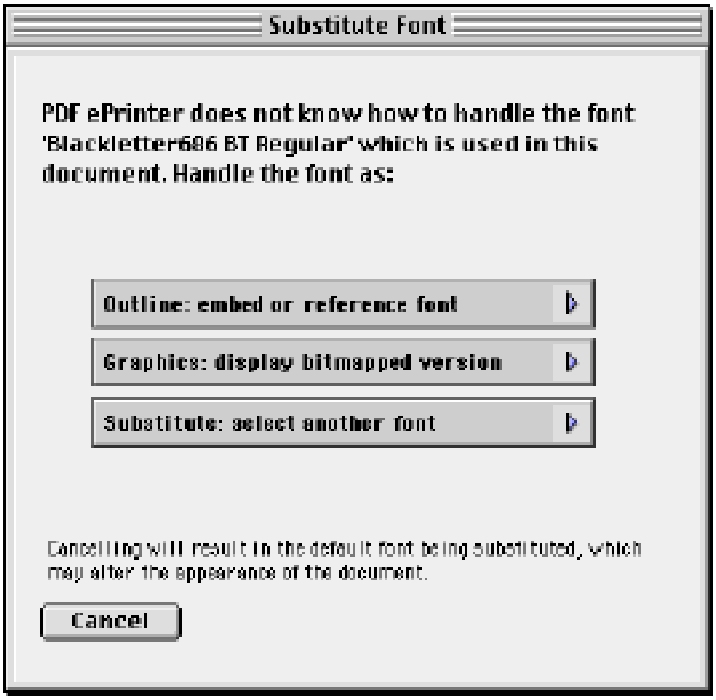
ClibPDF requires .PFB and .PFM files in order to embed fonts. The PDF Font Assistant will generate these as necessary, depending on what you already have:

If you have...	Font Assistant does this...
PFB & PFM	Adds the appropriate entry to the Fonts Manager
PFB & AFM	Converts the AFM to a PFM
LaserWriter Font & AFM	Converts the LW font to a PFB and the AFM to a PFM
TrueType	Generates a PFM, extracts the TrueType to a TTF, and generates a stub PFB for ClibPDF (this is swapped with the actual TTF during post-processing)

When preparing a Type 1 font, the Assistant attempts to guess the settings for the Bold and Italic checkboxes in the Fonts Manager. You should check these after the items have been added to the Fonts Manager list to ensure they are correct and make any changes as necessary.

In that this is a complex process, you should use the Font Assistant and only tweak entries manually if necessary. Also, the .PFM file generated should not be used with other applications; they are minimal files built purely to satisfy ClibPDF's requirements. The extracted TTF files should likewise not be used elsewhere.

If you are processing a document and an unmapped font is encountered, the following dialog is displayed if you have selected Manual font substitution mode in the Preferences Dialog:



Substitute Font Assistant

You can choose any of the three methods for handling a font from here and one of the windows discussed above will be opened for you to prepare the font for use with PDF ePrinter.

If you have selected the Suggest or Automatic font substitution modes, PDF ePrinter uses the following algorithm to suggest or choose a font:

1. If the font is a "recognized" font that is similar to one of the built-in fonts, the appropriate face of the built-in font is selected.

2. If the font is in the Japanese, Traditional or Simplified Chinese or Korean scripts, an appropriate built-in CJK font is selected.

3. If the font is a TrueType font, it is prepared for use. If you have selected to embed by default, and the font is less than about 100K in size, it is embedded. If it is greater than 100K or you have chosen not to embed, it is only referenced.

4. If none of the above provides a font, the default font selected in the Preferences dialog is used.

In Automatic mode, whatever font is selected above is used. If you have chosen to use saved font mappings, these selections become permanent. In Suggest mode, you are asked whether you want to use the suggested substitution. If you choose not to, the Substitute Font Assistant dialog above is displayed for you to select a method to handle the font.

Within the Fonts folder, PDF ePrinter maintains a file called ‘fontmap.lst’ which is used by ClibPDF to identify fonts. PDF ePrinter modifies this file with comments that contain information it requires to map these fonts. When you add a font using the PDF Fonts Manager, it updates and saves this file.

Multithreaded Operation

PDF ePrinter can process several jobs simultaneously. Normally, the user interface inhibits this because you can only manually prepare one job at a time. However, if you have a large job executing, you can begin the next job by using the Get Next Job command in the File menu (or by opening an existing spool file, etc.)[3] This will open a new job window in which you can prepare another job for processing.

This multithreading capability is most effectively used through AppleScript (or AppleEvents). Another application can use PDF ePrinter as a background processor to create PDF’s on the fly. Each job that is opened is assigned a unique job ID which is returned (or can be obtained) through the AppleScript interface. Use this job ID (an integer) to send commands to the job in order to prepare it for conversion. After you have initiated processing with the CreatePDF command, you can check for completion by using the JobExists command with the job’s job ID; when the job no longer exists, processing is complete and the PDF file is ready for use.

The most important issue for multithreaded operations is memory. You will need to allocate as many times more memory to the PDF ePrinter application as the number of jobs you expect to process. Of course, the amount of memory you need per job depends on the complexity; if you don’t have any graphics, you will need very little memory for each job. A document such as this manual with embedded graphics could require 4-6MB for each job session. Also, the memory warning feature does not work as well in the multithreaded environment, particularly since these types of operations are usually run unattended.

If you are having problems with processing multiple jobs, particularly machine freezes or sudden quits with no warnings, try increasing memory. Check for an stderr file in the PDF ePrinter folder. If it exists and there are any warnings about memory or "ZLIB," a lack of memory is probably the problem.

---

[3]This assumes that the *Exclusive processor use* option in the Preferences dialog is off; if it is on, *nothing* else will happen on the computer while a job is processing.

Troubleshooting

Converting a QuickDraw printer spool file to a PDF document is not an exact science. The QuickDraw graphics model and the PDF graphics model are fundamentally different; there are some capabilities in each that do not translate well to the other. In some cases, the output can be improved by tweaking the way in which the file is processed; PDF ePrinter provides numerous controls for doing this. However, it is possible that you will encounter a file which PDF ePrinter cannot fully or properly process.

The first step is to isolate the page that is causing the problem. You should be able to tell which page has the problem by either looking at the output PDF file (if the job processed to the end), or watching the progress window for the page number of the last processed page. Try reprinting just the suspect page. If that works, reprint and include the page prior to the suspect page. Keep adding prior pages until the problem recurs. In this case, it is probably a cumulative error and may be related to memory; try giving PDF ePrinter more memory or increasing the available objects on the Advanced Tab. Note: if you turn zlib compression off, this can greatly increase the size of the output file (this manual is approximately 200K compressed and over 5MB uncompressed) and will require substantially more memory to process.

You can have PDF ePrinter monitor memory during conversion; this can be set in the Preferences dialog. If the available heap memory gets below a set percentage of its initial value, PDF ePrinter will pause the conversion and warn you. You will be given the option of proceeding and cancelling the job. If you proceed, you most likely will crash if there are any further memory-intensive operations; you will not receive another warning. You should cancel the job, quit PDF ePrinter and assign it more memory in the Finder, and then try to print again. Note that this margin is dependent on the amount of memory you have allocated to PDF ePrinter. If you have a large amount allocated, and only a few pages left to convert, you may be able to complete the job. As you increase the memory allocated to PDF ePrinter, you can decrease the margin. A rule-of-thumb is to keep at least 250KB of memory as a buffer. To see the amount of memory PDF ePrinter has available initially, start it and go to the Preferences window. The current free heap and warning point are displayed near the bottom. The About window also displays memory usage.

If PDF ePrinter quits unexpectedly, check the PDF ePrinter folder for a file called "stderr." This file, if present, may contain an error message from ClibPDF that PDF ePrinter could not catch. The most common error you will find in this file is a message that you have exceeded the maximum number of images allowed in the PDF (although this condition will not cause the application to quit); increase the Max Images parameter on the Advanced tab (or increase the JPEG conversion cutoff to try to inline more images). You may also find information about problems with fonts, particularly if you are trying to use imported PFB/PFM files.

As stated above, some applications produce files which are not well suited for conversion to PDF through the method used by PDF ePrinter. Attempting to process these files with PDF ePrinter will result in poor-quality PDF. They may also cause PDF ePrinter to crash. If you find during processing that a large number of JPEG or inline images is being stored in the PDF file, the source file is probably not suitable for conversion.

Support

PDF ePrinter has been tested with several representative applications, but not all possible applications. If you find a problem with PDF ePrinter, please send a brief description to the author at mdsw@mac.com. Include the name of the application and any specifics you may have discovered in troubleshooting the problem. I can't guarantee I'll be able to solve the problem, but all reports will be reviewed. Regularly check the PDF ePrinter web site for software updates.

Acknowledgments

PDF ePrinter uses ClibPDF by FastIO Systems, Inc. It is incorporated and distributed under the terms of the license defined in section 5.2, "Private Non-Profit User Exemption." The unmodified source code for ClibPDF is available at http://www.fastio.com/

PDF ePrinter uses the zlib compression library written by Jean-loup Gailly and Mark Adler. This source code is available for free at ftp://ftp.freesoftware.com/pub/infozip/zlib/

Both of these C-language libraries were modified and compiled using Apple Computer, Inc.'s MPW environment and the MrC compiler, freely available at

http://developer.apple.com/tools/mpw-tools/

The PDF ePrinter application was written in RealBasic created by RealSoftware, Inc. It can be found at http://www.realbasic.com/

REALbasic: The Definitive Guide by Matt Neuburg was an invaluable resource for this project.

This manual was written using Nisus Writer, by Nisus Software, Inc., a free version of which can be found at http://www.nisus.com/free/LoginNisusWriter.asp

The installer for this product was created using Installer VISE Lite from MindVision Software. For more information on Installer VISE Lite, contact:

MindVision Software
5901 North 58th Street
Lincoln, NE 68507
Voice: (402) 323-6600
Fax: (402) 323-6611
E-mail: mindvision@mindvision.com
http://www.mindvision.com

This manual was converted to PDF using PDF ePrinter.